



SEGUNDO PROJETO DE ANÁLISE NUMÉRICA II

O Problema de Três Corpos Restrito

ALUNO: Fabricio C. Machado

RA: 102174

`fabcm1@gmail.com`

Junho de 2013

Resumo

Neste trabalho é discutido o Problema de Três Corpos e sua versão restrita, onde um corpo com massa desprezível é submetido à influência gravitacional de dois corpos com massa em movimento circular em torno de seu centro de massa. Após análise dos pontos de estabilidade (conhecidos como Pontos de Lagrange) são conduzidas simulações numéricas comparando-se o desempenho de solvers disponíveis no Matlab (baseados em métodos de passo variável) e o desempenho de um método de passos fixos.

Sumário

1	Introdução	3
1.1	O Problema de Três Corpos	3
1.2	O Problema de Três Corpos Restrito	3
2	Formulação Física	3
2.1	Sistema de Unidades	3
2.2	Referencial	4
2.3	Equações	4
3	Pontos de Lagrange	5
3.1	Potencial Generalizado	5
4	Simulações Numéricas	7
4.1	Erros	7
4.2	Simulações	7
4.3	Comparação entre diferentes solvers	7
5	Comparação com método de passos fixos	9
6	Conclusão	9
	Referências Bibliográficas	10
	Simulações no Matlab	11
S.1	Simulação 1	11
S.2	Simulação 2	12
S.3	Simulação 3	13
S.4	Simulação 4	14
S.5	Simulação 5	16
S.6	Potencial	18
S.7	Funções auxiliares	19

1 Introdução

1.1 O Problema de Três Corpos

O *Problema de Três Corpos* consiste em determinar o movimento de três corpos, dadas suas massas, posições e velocidades iniciais e considerando que estes corpos interagem gravitacionalmente entre si (isto é, com uma força de atração, central e proporcional às massas e ao inverso do quadrado da distância entre os corpos).

Este problema começou a ser estudado por Newton, em seu livro *Principia* e a partir de trabalhos de Heinrich Bruns (1887) e Henri Poincaré (1890) chegou-se à conclusão que o problema não pode ser resolvido em termos de expressões algébricas e integrais [1]. Entretanto, o problema ainda é objeto de pesquisa, sendo interessante determinar condições iniciais que levam a órbitas limitadas (onde um corpo não é ejetado do sistema) e com órbitas periódicas¹ (este tipo de trajetória é interessante, pois estas podem vir a ser encontradas na natureza, especialmente se forem estáveis).

Exemplos de ocorrência deste problema são a interação entre uma estrela, um planeta e seu satélite e a interação entre estrelas (sistemas com múltiplas estrelas são comuns na galáxia!)

1.2 O Problema de Três Corpos Restrito

O *Problema de Três Corpos Restrito* é uma simplificação do problema anterior introduzida por Euler. Neste caso, consideramos um terceiro corpo de massa desprezível, isto é, ele sofre influência gravitacional dos outros dois corpos, mas não interfere em seu movimento. Esta simplificação facilita a resolução do problema, dado que as soluções do problema de dois corpos é conhecida (os dois corpos orbitam seu centro de massa em órbitas elípticas, seguindo as leis de Kepler) e ainda assim é útil para modelar sistemas físicos, como por exemplo a interferência do Sol no movimento da Lua em torno da Terra e o movimento de uma nave espacial entre a Terra e a Lua.

2 Formulação Física

Neste trabalho, faremos simulações numéricas da órbita de uma nave em torno da Terra e da Lua. Para isto, definiremos um sistema de unidades e um referencial adequado para o problema.

2.1 Sistema de Unidades

1. A unidade de comprimento será a distância entre a Terra e a Lua: $d_{TL} := 1$
2. A unidade de massa será as somas das massas da Terra e da Lua: $M_T + M_L := 1$

¹Recentemente (março de 2013) foram descobertas 13 novos padrões de solução, elevando o número de padrões conhecidos para 16 [2]! Mais informações em <http://news.sciencemag.org/sciencenow/2013/03/physicists-discover-a-whopping.html>.

3. A unidade de tempo será tal que a velocidade angular de rotação da Terra e da Lua em torno de seu centro de massa seja igual a 1. Isto é, sendo T_{TL} o período de rotação do sistema, $\frac{T_{TL}}{2\pi} := 1$.

Neste sistema de unidades, a constante gravitacional G é igual a 1.

2.2 Referencial

Definimos como origem do referencial o centro de massa da Terra e da Lua e consideraremos ambos fixos no eixo x , de modo que suas coordenadas sejam $T = (-\mu, 0)$ e $L = (1 - \mu, 0)$. μ deve ser tal que $\mu.M_T = (1 - \mu).M_L \Rightarrow \mu = \frac{M_L}{M_T + M_L} \Rightarrow \mu = 0.012277471$ no caso da Terra e da Lua.

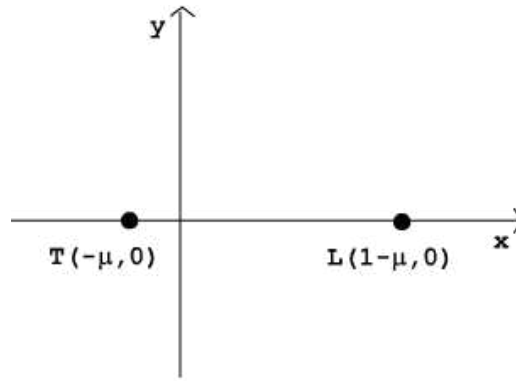


Figura 1: Coordenadas da Terra e da Lua no referencial adotado.

Para que as coordenadas da Terra e da Lua sejam fixas, o sistema de referência deve rotacionar junto com estes e portanto ser não inercial. Se $(\bar{x}, \bar{y})$ forem as coordenadas em um sistema inercial (onde a Terra e a Lua não são fixas, giram), a transformação para o referencial adotado será:

$$\begin{cases} \bar{x} = x \cdot \cos t - y \cdot \sin t \\ \bar{y} = x \cdot \sin t + y \cdot \cos t \end{cases}$$

2.3 Equações

Sendo $x(t)$ e $y(t)$ as coordenadas seguidas pela nave e denotando por $\dot{x}(t), \dot{y}(t)$ suas velocidades e $\ddot{x}(t), \ddot{y}(t)$ suas acelerações, o sistema de equações que descreve sua trajetória é:

$$\begin{pmatrix} \ddot{x} \\ \ddot{y} \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} - 2 \begin{pmatrix} -\dot{y} \\ \dot{x} \end{pmatrix} - \frac{\mu}{r_L^3} \begin{pmatrix} x - (1 - \mu) \\ y \end{pmatrix} - \frac{1 - \mu}{r_T^3} \begin{pmatrix} x + \mu \\ y \end{pmatrix}$$

$$r_L = \sqrt{(x - (1 - \mu))^2 + y^2}$$

$$r_T = \sqrt{(x + \mu)^2 + y^2}$$

Se multiplicarmos ambos os lados pela massa da nave, podemos interpretar a equação como a segunda lei de Newton, com cada termo correspondendo a uma força que age sobre a nave.

Os dois primeiros termos são consequência do referencial não inercial adotado. O primeiro termo é a força centrífuga, o segundo a força de Coriolis (que é sempre ortogonal ao movimento do satélite). Os outros dois termos correspondem à força gravitacional exercida pela Lua e pela Terra, respectivamente.

3 Pontos de Lagrange

Os pontos de Lagrange são pontos de equilíbrio do sistema, isto é, condições iniciais que geram soluções estacionárias. Além de $\dot{x}(0) = \dot{y}(0) = 0$, devemos ter $(x(0), y(0))$ tais que $\ddot{x}(0) = \ddot{y}(0) = 0$ segundo as equações apresentadas na sessão anterior.

O sistema é não conservativo, pois possui uma força que depende da velocidade da partícula (força de Coriolis). Isto significa que não é possível definir um campo potencial que represente as forças no sistema. Ainda assim, podemos considerar um *potencial generalizado*, baseado apenas nas forças que não dependem da velocidade (afinal, para a análise dos pontos de equilíbrio, podemos considerar a velocidade nula). Este sistema será conservativo.

3.1 Potencial Generalizado

O potencial do sistema pode ser calculado como a soma dos potenciais de cada uma das forças. No caso da força gravitacional, temos $U_{GT} = \frac{\mu-1}{r_T}$ e $U_{GL} = \frac{-\mu}{r_L}$ e no caso da força centrífuga, temos $U_C = -\frac{r^2}{2}$.

O sistema considerado neste trabalho possui exatamente 5 pontos de equilíbrio, denominados $L1, L2, L3, L4$ e $L5$. Podemos localizar estes pontos como pontos críticos do campo potencial (isto é, pontos nos quais $\nabla U = 0$). Estes pontos estão indicados na figura 3.

Em todos os 5 pontos, a resultante da força gravitacional da Terra e da Lua deve ser tal que compense a força centrífuga (no referencial não inercial). Assim, o satélite poderá orbitar em torno do centro de massa da Terra e da Lua com a mesma velocidade angular que o sistema.

Os pontos $L1, L2$ e $L3$ são instáveis (isto é, um objeto posicionado próximo do ponto será empurrado para longe), ainda assim, no sistema Terra-Sol, os pontos $L1$ e $L2$ são importantes para o posicionamento de satélites (um satélite pode permanecer lá através de pequenos propulsores para corrigir sua posição), como o observatório solar SOHO em $L1$.

É interessante notar que a estabilidade dos pontos $L4$ e $L5$ não segue diretamente da análise do potencial, já que a força de Coriolis influencia nesta questão. De fato, apesar de serem pontos de máximo local do potencial generalizado, surpreendentemente, podem ser pontos estáveis²! Os asteróides troianos são exemplos de corpos que se localizam nos pontos $L4$ e $L5$ do sistema Sol-Júpiter.

²Mais informações, incluindo uma análise sobre a estabilidade dos pontos, pode ser encontrada em [5].

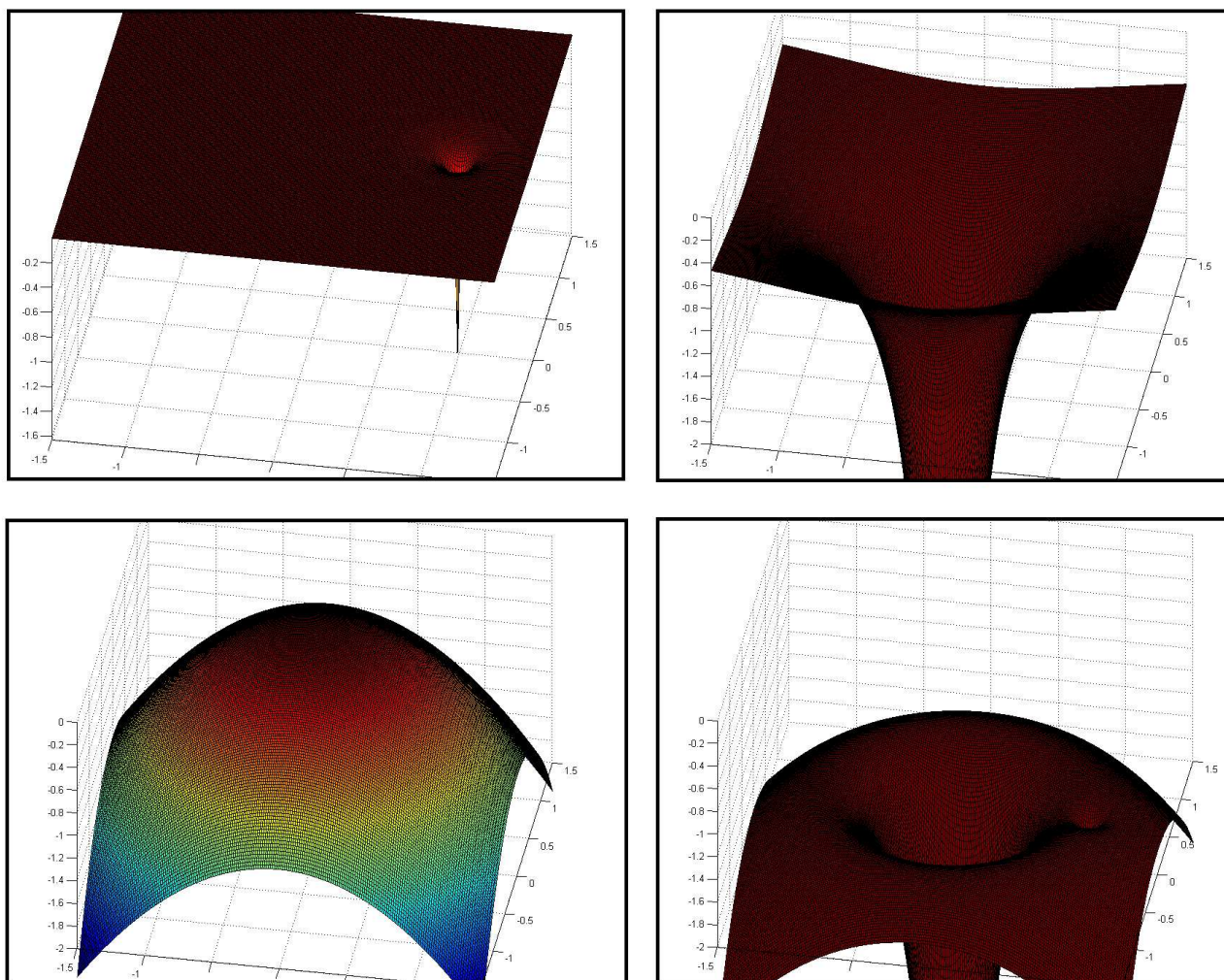


Figura 2: Respectivamente, os potenciais induzidos pelos campos da Lua, da Terra e pela força centrífuga. No canto inferior direito, o potencial resultante.

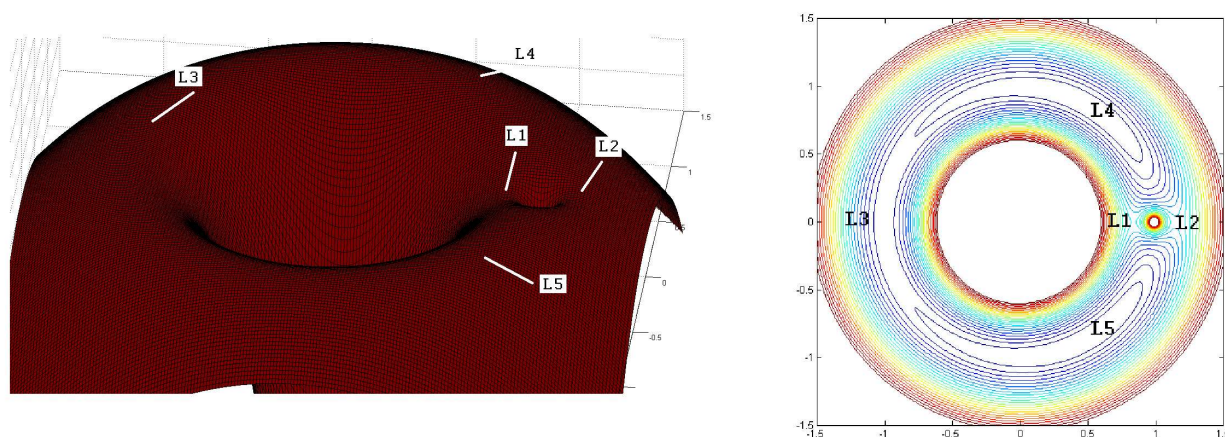


Figura 3: Os 5 pontos de Lagrange. À direita, o potencial visto através de curvas de nível.

4 Simulações Numéricas

4.1 Erros

O sistema foi simulado com o método `ode45` do `Matlab`, que é um método de passos adaptativos, que ajusta o tamanho de cada passo de forma a limitar o erro de truncamento local $|e(k)|$ em cada coordenada:

$$|e(k)| \leq \max(RelTol * abs(y(k)), AbsTol)$$

Na simulação, foram usados os valores $RelTol = 10^{-3}$ e $AbsTol = 10^{-8}$. $RelTol$ expressa a quantidade de algarismos significativos a ser levada em conta, já $AbsTol$ representa a precisão absoluta máxima, que é usada no lugar da precisão relativa quando o valor absoluto de $y(k)$ é muito pequeno.

4.2 Simulações

A primeira simulação foi realizada com as condições iniciais:

$$y_0 = (0.994, 0, 0, -2.00159) \text{ (A)}$$

Sendo que a terceira e quarta coordenadas se referem às velocidades $\dot{x}(0)$ e $\dot{y}(0)$, respectivamente. Estas condições descrevem uma nave inicialmente à direita da Lua e lançada para baixo. Na figura 4, a trajetória percorrida pela nave.

Em azul vemos os pontos realmente calculados pelo método. A trajetória desenhada foi obtida via interpolação a partir destes pontos. Observamos que nas proximidades da Lua foram calculados mais pontos, consequência da necessidade de se tomar um passo menor para manter o controle do erro nessa região.

A simulação seguinte foi realizada com condições iniciais ligeiramente alteradas:

$$y_0 = (0.994, 0, 0, -2.0317) \text{ (B)}$$

Observa-se que neste caso a trajetória foi bem diferente, o que demonstra a instabilidade do sistema em relação às condições iniciais e a necessidade de um método com controle de erro para a simulação.

4.3 Comparação entre diferentes solvers

Foram realizadas novas simulações com as condições iniciais (A) da primeira simulação, comparando-se os resultados produzidos por diversos solvers disponíveis no `Matlab`. O solver padrão `ode45`, um método de passos variáveis baseado em métodos de Runge-Kutta de ordem 4 e 5 (par de Dormand-Prince). `ode23`, um método de passos variáveis baseado em métodos de Runge-Kutta de ordem 2 e

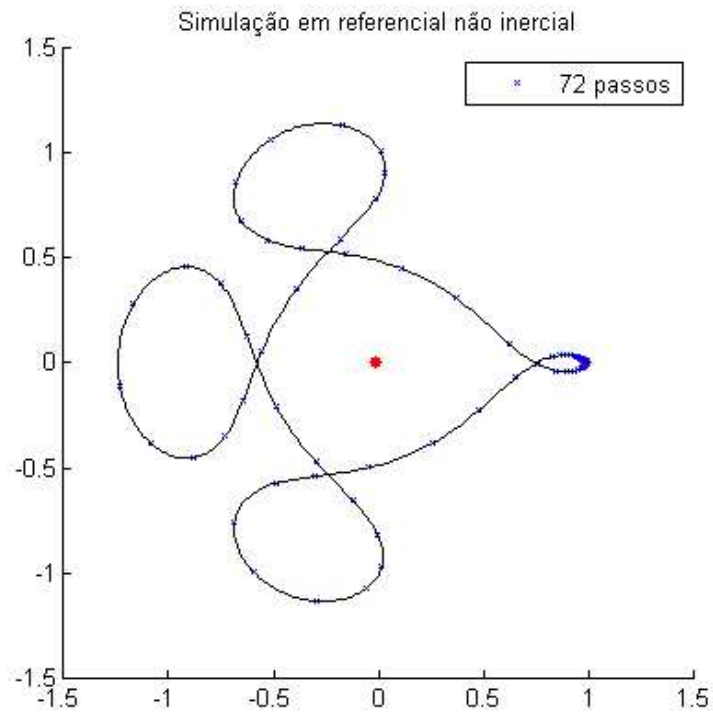


Figura 4: Trajetória da nave na primeira simulação.

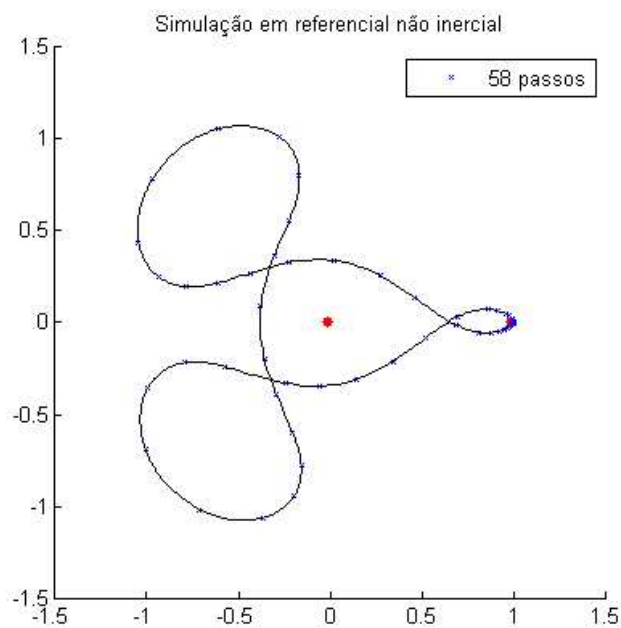


Figura 5: Trajetória da nave na segunda simulação.

3 (par de Bogacki e Shampine). *ode113*, com uma implementação do método de Adams-Bashforth-Moulton de ordem variável e com passos múltiplos (usa valores calculados em iterações anteriores para efetuar a próxima iteração).

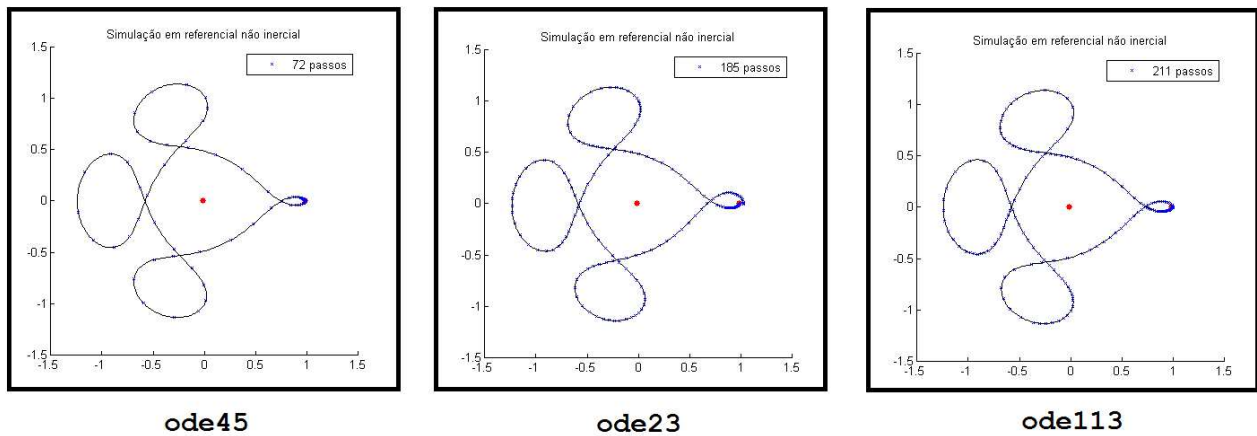


Figura 6: Trajetórias calculadas pelos solvers.

	ode45	ode23	ode113
n° passos	72	185	211
n° avaliações	517	613	438
tempo (s)	0.061676	0.071082	0.205601

Comparação entre diferentes solvers para a simulação do mesmo PVI.

Todos os solvers resolveram o problema em pouco tempo. Observamos que *ode45* efetuou menos passos (consequência de ser um método de maior ordem) e *ode113* efetuou menos avaliações de função (consequência de ser um método de passos múltiplos).

5 Comparação com método de passos fixos

Para comparação com os resultados obtidos pelo método de passo adaptativo, foi implementado o método de Runge-Kutta de ordem 4 clássico, com passos fixos.

A seguir, as órbitas obtidas na simulação com as condições iniciais (A) e diversos números de passos.

Observamos a necessidade de um grande número de passos para se obter resultados semelhantes aos obtidos pelo método de passos variáveis. Isto é consequência da necessidade de passos curtos para se manter controle do erro na região próxima da Lua e da possibilidade de passos mais longos nas outras regiões.

6 Conclusão

Neste trabalho foi possível observar a estreita relação entre as condições iniciais e as trajetórias percorridas pela solução de certos problemas de valor inicial e a necessidade do uso dos métodos de

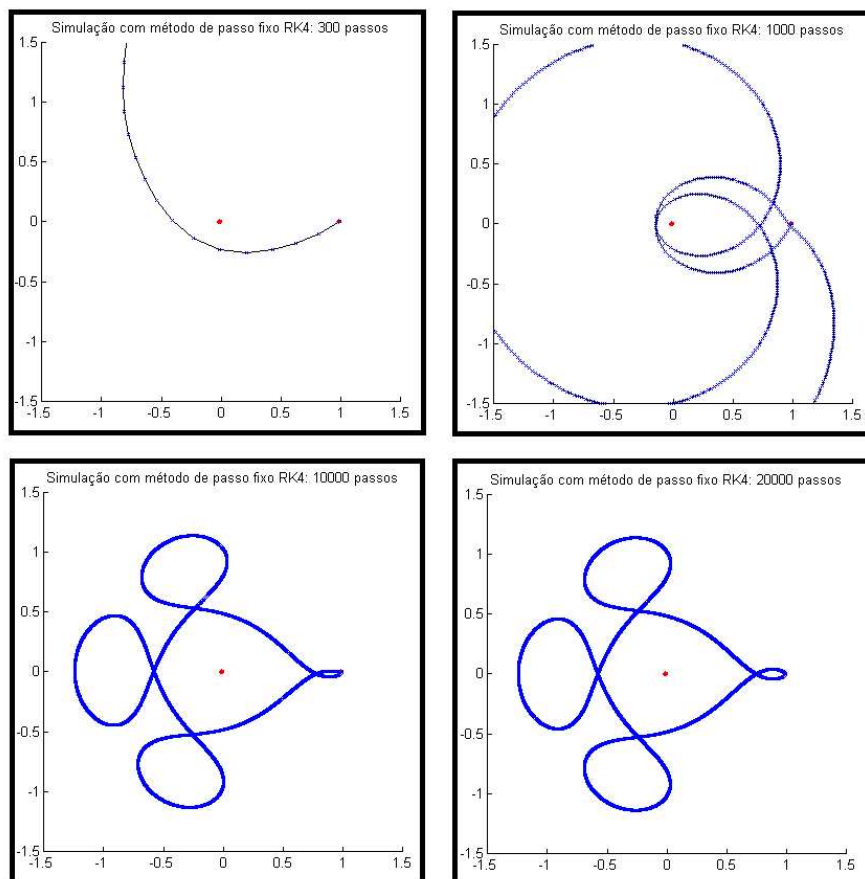


Figura 7: Simulações feitas com 300, 1000, 10000 e 20000 passos.

passo variável para a simulação de sistemas que se apresentam instáveis em regiões específicas.

Referências Bibliográficas

- [1] WOLFRAM, Stephen *A new kind of science* Champaign, IL: Wolfram Media, c2002.
- [2] Šuvakov, Milovan and Dmitrašinović, V. *Three Classes of Newtonian Three-Body Planar Periodic Orbits*, Phys. Rev. Lett. 110, 114301 (2013)
- [3] U. M. Ascher and L. R. Petzold *Computer methods for ordinary differential equations and differential-algebraic equation*, Philadelphia: SIAM, 1998.
- [4] C. H. Edwards Jr. and D. E. Penney, *Equações Diferenciais Elementares com Problemas de Contorno*, 3ed. Rio de Janeiro: Prentice Hall do Brasil, 1995.
- [5] Cornish, Neil J., *The Lagrangian Points*, Montana State University - Department of Physics. Retrieved 29 July 2011. Disponível em: <http://www.physics.montana.edu/faculty/cornish/lagrange.pdf>

```

% ***** SIMULAÇÃO 1 *****
%- PVI visto em referencial inercial por 3 ciclos e meio
clear;
close;

% unidade de distancia: d(M,E) = 384400 km
% unidade de tempo: 27.32 / 2pi = 4.35 dias
% unidade de massa: mM + ME

% Parametros do sistema
ni = 0.012277471; % mM / (mE + mM)
E = [-ni, 0];
M = [1- ni, 0];
bodies = [E;M]; %(0,0) é o centro de massa do sistema

y0 = [ 0.994, 0, 0, -2.0015851]; % Condições iniciais
tspan = linspace(0, 39, 800); % Intervalo de simulação
option1 = odeset('RelTol', 1e-6, 'AbsTol', 1e-8);
[T1,Y1] = ode45(@f, tspan , y0, option1); % Resolve PVI
%Obs: Apesar do solver retornar valores da solução em pontos igualmente
%espaçados, a solução foi calculada em pontos com passo variável. Os pontos
%exibidos são obtidos via interpolação.

% Desenha objetos em suas posições iniciais e define propriedades da janela
% de exibição.
hold on;
[eg1, eg2] = drawcircle(0.0166, bodies(1,1), bodies(1,2));
[mg1, mg2] = drawcircle(0.0045, bodies(2,1), bodies(2,2));
sat_graphic = plot(y0(1), y0(2), 'k');
xlim([-1.5,1.5]); ylim([-1.5,1.5]); axis equal; xlim('manual');
title('Simulação em referencial inercial');
YR = zeros(1,2);

for k = 1 : length(Y1)
    %Rotaciona os corpos para voltar ao referencial inercial.
    t = T1(k);
    bodiesrot = bodies * [cos(t), sin(t); -sin(t), cos(t)];
    delete(eg1); delete(mg1); delete(eg2); delete(mg2);
    [eg1, eg2] = drawcircle(0.0166, bodiesrot(1,1), bodiesrot(1,2));
    [mg1, mg2] = drawcircle(0.0045, bodiesrot(2,1), bodiesrot(2,2));

    YR(1:2) = Y1(k,1:2) * [cos(t), sin(t); -sin(t), cos(t)];

    delete (sat_graphic);
    sat_graphic = plot(YR(1), YR(2), 'k.', 'MarkerSize',20);
    plot(YR(1), YR(2), 'k.', 'MarkerSize',2);

    pause(0.002);
end

```

```

% ***** SIMULAÇÃO 2 *****
%- PVI visto em referencial não inercial por 1 ciclo
clear;
close;

% unidade de distancia: d(M,E) = 384400 km
% unidade de tempo: 27.32 / 2pi = 4.35 dias
% unidade de massa: mM + ME

% Parametros do sistema
ni = 0.012277471; % mM / (mE + mM)
E = [-ni, 0];
M = [1- ni, 0];
bodies = [E;M]; %(0,0) é o centro de massa do sistema

% Resolve o PVI exibindo apenas os pontos realmente calculados pelo solver
y0 = [ 0.994, 0, 0, -2.00159]; % Condições iniciais
option = odeset('RelTol', 1e-3, 'AbsTol', 1e-8, 'Events', @myevents, 'Refine', 1,
'Stats', 'on');
tic
[Tit, Yit] = ode45(@f, [0, 39], y0, option);
toc

option = odeset(option, 'Stats', 'off');
tfinal = Tit(length(Tit));
tspan = linspace(0, tfinal, 500);
[T,Y] = ode45(@f, tspan , y0, option);

hold on;
[eg1, eg2] = drawcircle(0.0166, bodies(1,1), bodies(1,2));
[mg1, mg2] = drawcircle(0.0045, bodies(2,1), bodies(2,2));
sat_graphic = plot(y0(1), y0(2), 'k');
xlim([-1.5,1.5]); ylim([-1.5,1.5]); axis equal; xlim('manual');
title('Simulação em referencial não inercial');

j = 1;
for k = 1 : length(Y) - 1
    plot(Y(k:k+1,1), Y(k:k+1,2), 'k-');

    %desenho dos pontos obtidos pelo solver
    jf = j;
    while Tit(jf) < T(k+1)
        jf = jf + 1;
    end
    it = plot(Yit(j:jf,1), Yit(j:jf,2), 'xb ', 'MarkerSize',3);
    j = jf;

    pause(0.002);
end
legend([it], sprintf('%d passos', length(Tit)-1));

```

```

% ***** SIMULAÇÃO 3 *****
%- PVI visto em referencial não inercial por 1 ciclo, condições iniciais
%ligeiramente diferentes da simulação 2
clear;
close;

% unidade de distancia: d(M,E) = 384400 km
% unidade de tempo: 27.32 / 2pi = 4.35 dias
% unidade de massa: mM + ME

% Parametros do sistema
ni = 0.012277471; % mM / (mE + mM)
E = [-ni, 0];
M = [1- ni, 0];
bodies = [E;M]; %(0,0) é o centro de massa do sistema

% Resolve o PVI exibindo apenas os pontos realmente calculados pelo solver
y0 = [ 0.994, 0, 0, -2.0317 ]; % Condições iniciais
option = odeset('RelTol', 1e-3, 'AbsTol', 1e-8, 'Events', @myevents, 'Refine', 1);
[Tit, Yit] = ode45(@f, [0, 39], y0, option);

tfinal = Tit(length(Tit));
tspan = linspace(0, tfinal, 500);
[T,Y] = ode45(@f, tspan , y0, option);

hold on;
[eg1, eg2] = drawcircle(0.0166, bodies(1,1), bodies(1,2));
[mg1, mg2] = drawcircle(0.0045, bodies(2,1), bodies(2,2));
sat_graphic = plot(y0(1), y0(2), 'k');
xlim([-1.5,1.5]); ylim([-1.5,1.5]); axis equal; xlim('manual');
title('Simulação em referencial não inercial');

j = 1;
for k = 1 : length(Y) - 1
    plot(Y(k:k+1,1), Y(k:k+1,2), 'k-');

    %desenho dos pontos obtidos pelo solver
    jf = j;
    while Tit(jf) < T(k+1)
        jf = jf + 1;
    end
    it = plot(Yit(j:jf,1), Yit(j:jf,2), 'xb ', 'MarkerSize',3);
    j = jf;

    pause(0.002);
end
legend(it, sprintf('%d passos', length(Tit)-1));

```

```
% ***** SIMULAÇÃO 4 *****
%- PVI visto em referencial não inercial por 1 ciclo e simulado com metodo
%de passo fixo
clear;
close;

% unidade de distancia: d(M,E) = 384400 km
% unidade de tempo: 27.32 / 2pi = 4.35 dias
% unidade de massa: mM + ME

% Parametros do sistema
ni = 0.012277471; % mM / (mE + mM)
E = [-ni, 0];
M = [1- ni, 0];
bodies = [E;M]; %(0,0) é o centro de massa do sistema

% Resolve o PVI exibindo apenas os pontos realmente calculados pelo solver
y0 = [ 0.994, 0, 0, -2.00159]; % Condições iniciais
[T1, Y1] = RK4(@f, [0, 17.0584], y0, 300);
[T2, Y2] = RK4(@f, [0, 17.0584], y0, 1000);
[T3, Y3] = RK4(@f, [0, 17.0584], y0, 10000);
[T4, Y4] = RK4(@f, [0, 17.0584], y0, 20000);

% subplot(2,2,1);
hold on;
drawcircle(0.0166, bodies(1,1), bodies(1,2));
drawcircle(0.0045, bodies(2,1), bodies(2,2));
plot(y0(1), y0(2), 'k');
xlim([-1.5,1.5]); ylim([-1.5,1.5]); axis equal; xlim('manual');
title('Simulação com método de passo fixo RK4: 300 passos');
plot(Y1(:,1), Y1(:,2), 'k-');
plot(Y1(:,1), Y1(:,2), 'xb ', 'MarkerSize',3);

pause;
close;
% subplot(2,2,2);
hold on;
drawcircle(0.0166, bodies(1,1), bodies(1,2));
drawcircle(0.0045, bodies(2,1), bodies(2,2));
plot(y0(1), y0(2), 'k');
xlim([-1.5,1.5]); ylim([-1.5,1.5]); axis equal; xlim('manual');
title('Simulação com método de passo fixo RK4: 1000 passos');
plot(Y2(:,1), Y2(:,2), 'k-');
plot(Y2(:,1), Y2(:,2), 'xb ', 'MarkerSize',3);

pause;
close;
% subplot(2,2,3);
hold on;
drawcircle(0.0166, bodies(1,1), bodies(1,2));
drawcircle(0.0045, bodies(2,1), bodies(2,2));
plot(y0(1), y0(2), 'k');
xlim([-1.5,1.5]); ylim([-1.5,1.5]); axis equal; xlim('manual');
title('Simulação com método de passo fixo RK4: 10000 passos');
plot(Y3(:,1), Y3(:,2), 'k-');
```

```
plot(Y3(:,1), Y3(:,2), 'xb ', 'MarkerSize',3);

pause;
close;
% subplot(2,2,4);
hold on;
drawcircle(0.0166, bodies(1,1), bodies(1,2));
drawcircle(0.0045, bodies(2,1), bodies(2,2));
plot(y0(1), y0(2), 'k');
xlim([-1.5,1.5]); ylim([-1.5,1.5]); axis equal; xlim('manual');
title('Simulação com método de passo fixo RK4: 20000 passos');
plot(Y4(:,1), Y4(:,2), 'k-');
plot(Y4(:,1), Y4(:,2), 'xb ', 'MarkerSize',3);
```



```

% ***** SIMULAÇÃO 5 *****
%- Simulação de uma nave inicialmente próxima do ponto L5
clear;
close;

% unidade de distancia: d(M,E) = 384400 km
% unidade de tempo: 27.32 / 2pi = 4.35 dias
% unidade de massa: mM + ME

% Parametros do sistema
ni = 0.012277471; % mM / (mE + mM)
E = [-ni, 0];
M = [1- ni, 0];
bodies = [E;M]; % (0,0) é o centro de massa do sistema
L = E + (M - E)*[0.5, -sqrt(3)/2; sqrt(3)/2, 0.5];

% Resolve o PVI exibindo apenas os pontos realmente calculados pelo solver
% y0 = [ L(1) + 0.01, L(2), 0, 0 ]; % Condições iniciais
y0 = [ 0.45, -0.9, 0, 0 ];
option = odeset('RelTol', 1e-3, 'AbsTol', 1e-8, 'Events', @myevents, 'Refine', 1,
'Stats', 'on');
tic
[Tit, Yit] = ode45(@f, [0, 39], y0, option);
toc

option = odeset(option, 'Stats', 'off');
tfinal = Tit(length(Tit));
tspan = linspace(0, tfinal, 500);
[T,Y] = ode45(@f, tspan , y0, option);

%Cálculo do potencial
[x, y] = meshgrid(linspace(-1.5, 1.5, 200));
u = -ni./sqrt( (x - 1 + ni).^2 + y.^2 ) - ...
      (1-ni)./sqrt( (x + ni).^2 + y.^2 ) - (x.^2+ y.^2)/2;
contour(x,y,log(log(-u)), linspace(-1, -.5, 30));

hold on;
[eg1, eg2] = drawcircle(0.0166, bodies(1,1), bodies(1,2));
[mg1, mg2] = drawcircle(0.0045, bodies(2,1), bodies(2,2));
drawcircle(0.0045, L(1), L(2));
sat_graphic = plot(y0(1), y0(2), 'k');
xlim([-1.5,1.5]); ylim([-1.5,1.5]); axis equal; xlim('manual');
title('Simulação em referencial não inercial');

j = 1;
for k = 1 : length(Y) - 1
    plot(Y(k:k+1,1), Y(k:k+1,2), 'k-');

    %desenho dos pontos obtidos pelo solver
    jf = j;
    while Tit(jf) < T(k+1)
        jf = jf + 1;
    end
    it = plot(Yit(j:jf,1), Yit(j:jf,2), 'xb ', 'MarkerSize',3);

```

```
j = jf;  
  
    pause(0.002);  
end  
legend(it, sprintf('%d passos', length(Tit)-1));
```

% Desenha o potencial generalizado do Problema de Três Corpos Restrito

```
clear
close
ni = 0.012277471;
[x, y] = meshgrid(linspace(-1.5, 1.5, 200));
ugl = -ni./sqrt( (x - 1 + ni).^2 + y.^2 );
ugt = -(1-ni)./sqrt( (x + ni).^2 + y.^2 );
uc = - (x.^2+ y.^2)/2;
u = uc+ugt+ugl;
surf(x,y,u);
% contour(x,y,log(log(-u)), linspace(-1, -.5, 30));
axis equal;
zlim([-2,0]);
view(10,34);
zoom(1.3);
```

```
function dy = f(~, y)
%Função com as equações que descrevem o problema.

ni = 0.012277471; % mM / (mE + mM)

rE = sqrt( (y(1) + ni).^2 + y(2).^2 );
rM = sqrt( (y(1) -1 + ni).^2 + y(2).^2 );

dy = zeros(4,1);
dy(1) = y(3);
dy(2) = y(4);
dy(3) = y(1) + 2*y(4) - (1-ni)*(y(1)+ni)./rE.^3 - ni*(y(1)-1+ni)./rM.^3;
dy(4) = y(2) - 2*y(3) - (1-ni)*y(2)./rE.^3 - ni*y(2)./rM.^3;
```

```
end
```

```
function [T, Y] = RK4(f, tspan, y0, n)
%Implementação de um método Runge Kutta de 4 ordem de passos fixos.
```

```
T = linspace( tspan(1), tspan(2), n);
h = T(2) - T(1);
```

```
Y = zeros(n, length(y0));
Y(1,:) = y0;
```

```
for k = 1 : n-1
    k1 = f(T(k),Y(k,:));
    k2 = f(T(k) + h/2, Y(k,:) + h/2*k1');
    k3 = f(T(k) + h/2, Y(k,:) + h/2*k2');
    k4 = f(T(k) + h, Y(k,:) + h*k3');

    Y(k+1,:) = Y(k,:) + h/6*(k1' + 2*k2' + 2*k3' + k4');
end
```

```
function [b1, b2] = drawcircle(r, cx, cy)
%Função auxiliar usada para desenhar circulos
t = linspace(0, 2*pi, 20);
x = cx + r*cos(t);
y = cy + r*sin(t);
b1 = plot(x,y, '-r');
b2 = plot(cx,cy, '.r', 'MarkerSize',15);
```

```
end
```

```
function [value, isterminal, direction] = myevents(~,y)
%Função auxiliar usada pelo solver, para determinar o final da simulação.
```

```
[value, ~] = cart2pol(y(1), y(2));
if y(1) > 0
    isterminal = 1;
else
    isterminal = 0;
end
direction = -1;
```

```
end
```