

## 5.4 Movimento devido a forças centrais

### 5.4.1 O problema dos N-corpos

Nesta seção consideraremos o problema de três e mais corpos interagindo mutuamente via forças gravitacionais.

Em 1894, Poincaré demonstrou teoricamente a impossibilidade de determinar soluções gerais quando  $N \geq 3$ . Essa impossibilidade abriu caminho para o desenvolvimento de várias abordagens numéricas. O problema consta, matematicamente, na solução de  $N$  equações diferenciais vetoriais de segundo ordem, tendo em conta a interação de todos com todos.

Uma solução numérica deste problema pode ser obtida com praticamente qualquer exatidão. (Contudo, sob suposições especiais a respeito do tipo de movimento e interação, soluções analíticas para o problema de  $N$  corpos podem ser encontradas.)

O nosso estudo vai ser puramente numérico, salvo unas considerações teóricas no caso do "problema restrito", veja 5.4.5

A abordagem numérica nos oferece rapidamente soluções gráficas de problemas físicas complicadas. É só necessária olhar uma vez nas figuras representadas nesta seção. Na figura a seguir vemos 4 corpos interagindo via forças gravitacionais.

O nosso problema reza: Quais são as posições e velocidades dos 4 corpos no instante  $t$ , se estes dados são conhecidos no instante  $t = 0$ ?

A força gravitacional que o corpo- $k$  exerce sobre o corpo- $i$  é dada pela equação (1)

$$\vec{F}_{ik} = -G \frac{m_i m_k}{r_{ik}^3} \vec{r}_{ik}, \quad (1)$$

onde  $\vec{r}_{ik} = \vec{r}_i - \vec{r}_k = (x_i - x_k)\mathbf{i}_o + (y_i - y_k)\mathbf{j}_o + (z_i - z_k)\mathbf{k}_o$

e  $r_{ik} = ((x_i - x_k)^2 + (y_i - y_k)^2 + (z_i - z_k)^2)^{1/2}$ .

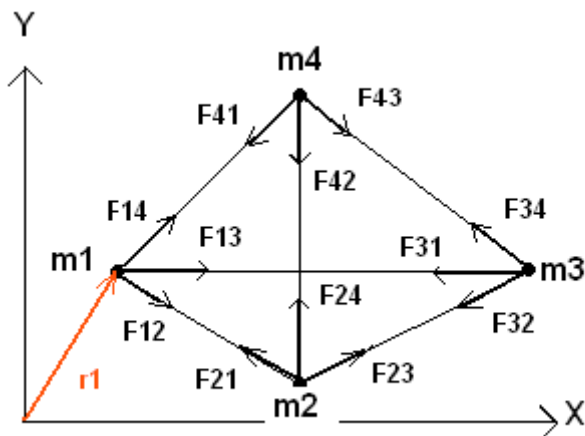


Fig.5.4-1

Para a força total sobre o corpo-i temos (na figura com  $N = 4$ )

$$\bar{F}_i = \sum_{k=1, k \neq i}^N \bar{F}_{ik} = -Gm_i \sum_{k=1, k \neq i}^N \frac{m_k}{r_{ik}^3} \bar{r}_{ik} \quad (2)$$

A soma dupla podemos substituir por somas simples como

$$m_i \frac{dv_{ix}}{dt} = \sum_{j=1}^N - \frac{Gm_i m_j (x_i - x_j)}{r_{ij}^3}$$

e duas outras para  $dv_{iy}/dt$  e  $dv_{iz}/dt$ . A soma não deve-se fazer para  $j = i$ . No seguinte programa utilizamos somente as coordenadas-x,y.

Nossa tarefa consiste em calcular primeiro as posições e velocidades dos 4 corpos para o final do intervalo  $h := \Delta t$ , supondo que estes valores sejam conhecidos ao início intervalo.

Começemos com o cálculo da velocidade  $v(t + h)$ . (Para simplificar, anotamos só as equações para a coordenada-x, mas sem escrever o índice x.)

Para o primeiro corpo temos

$$v_1(t+h) = v_1(t) + a_1(t)h = v_1(t) + F_1 h/m_1 \quad (3)$$

Lembramo-nos que  $a(t) \approx (v(t+h) - v(t))/h$ .

Da figura 5.4-1 vemos que para as coordenadas-x vale  $F_1 = F_{12} + F_{13} + F_{14}$  e a equação (3) deve ser escrita da seguinte maneira pormenorizada:

$$v_1(t+h) = v_1(t) + F_{12} h/m_1 + F_{13} h/m_1 + F_{14} h/m_1 \quad (4)$$

Para o cálculo numérico decompos a última equação em 3 equações da mesma estrutura:

$$v_1(t+h) = v_1^{(2)} + F_{14} h/m_1,$$

$$v_1^{(2)} = v_1^{(1)} + F_{13} h/m_1,$$

$$v_1^{(1)} = v_1 + F_{12} h/m_1.$$

Então temos que avaliar três vezes uma equação da forma

$$v^{\text{ novo}} = v^{\text{ velho}} + \text{ termo de correção} \quad (5)$$

Escrevendo  $R3:=R^3$  e  $R:=(x^2 + y^2)^{1/2}$  podemos formular para o primeiro corpo o seguinte esquema de cálculo:

$$\begin{aligned} 1 \rightarrow 2 \quad & x = x(1) - x(2) \\ & F = -GM(1)M(2) x/R^3 \\ & v(1) = v(1) + F \cdot h/M(1) := v_1^{(1)} \end{aligned}$$

$$\begin{aligned} 1 \rightarrow 3 \quad & x = x(1) - x(3) \\ & F = -GM(1)M(3) x/R^3 \\ & v(1) = v(1) + F \cdot h/M(1) := v_1^{(2)} \end{aligned}$$

$$\begin{aligned} 1 \rightarrow 4 \quad & x = x(1) - x(4) \\ & F = -GM(1)M(4) x/R^3 \\ & v(1) = v(1) + F \cdot h/M(1) := v_1(t + h) \end{aligned}$$

Para os corpos 2, 3 e 4 temos que escrever esquemas da mesma estrutura. Por exemplo, no caso do segundo corpo vamos tomar em conta que  $F_{12} = -F_{21}$ . Isso significa que na interação  $1 \rightarrow 2$  temos que incluir a equação  $v(2) = v(2) - F \cdot h/M(2)$ .

Igualmente incluímos a equação  $v(3) = v(3) - F \cdot h/M(3)$  ao calcular a interação  $1 \rightarrow 3$ . Nesta equação, F significa  $F_{13}$ , etc.

No caso do quarto corpo, não temos que calcular nenhuma força, pois todas as interações já foram tomadas em consideração.

Com as seguintes linhas de programa podemos fazer todos estes cálculos:

```

for i from 1 to n-1 do
  for k from i+1 to n do
    x:=x(i)-x(k):
    y:=y(i)-y(k):
    r:=sqrt(x^2 + y^2):
    r3:= r^3:
    F:= -g·m(i)·m(k)·x/r3:
    v(i):= v(i) + F·h/m(i):
    v(k):= v(k) - F·h/m(k):
  end_for:
end_for:

```

Com as seguintes linhas calculamos as coordenadas x e y (não usamos z)

```

for j from 1 to n do
  x(j) := x(j) + v(j)·h:
  y(j) := y(j) + v(j)·h:
end_for:

```

Sabe-se que a exatidão do método numérico será melhorada se no primeiro passo se calcula  $v$  no centro do intervalo (método de Feynman):  $v(t + h/2)$ . É por isso que, ao princípio, fazemos os cálculos com  $h/2$ . Após haver calculada a primeira velocidade com  $h/2$ , vamos usar sempre  $\Delta t = h$ .

Conforme expliquei no parágrafo 5.2.1, fazemos os cálculos com variáveis sem unidades (valores reduzidos). Com o primeiro programa calculamos o movimento de três corpos de igual massa que encontram-se ao começo nos vértices de um triângulo equilátero  $A(1|0)$ ,  $B(0|1,732)$ ,  $C(-1|0)$ . Dando-lhes ao principio as velocidades indicadas no programa, vamos observar que os corpos permaneceram descrevendo o mesmo círculo.

Variando as velocidades iniciais, produzirá trajetórias bem exóticas.

Use, p. ex., as velocidades  $(0.4,0.6)$ ,  $(0.3,-0.6)$  e  $(-0.7,0)$ . O resultado serão trajetórias caóticas. Mostra isso que não é fácil criar um mundo estável, precisa-se imaginação e muito tempo.

## Programa 1:

- ```

reset()://n-corpos com método de Feynman
N_corpos:=proc(dt,passos)// dt = intervalo do tempo

begin

m:=[1,1,1]://lista das massas
X:=[1,-1,0]:Y:=[0,0,1.732]:// coordenadas
U:=[.35355,.35355,-.7071]://coord.-x da velocidade
V:=[.61237,-.61237,0]:
n:=nops(m)//número de elementos na lista m

for p from 1 to n do
  x[p]:=X[p]://tira os elementos das listas
  y[p]:=Y[p]:
  u[p]:=U[p]:
  v[p]:=V[p]:
end_for:

x1[0]:=X[1]:y1[0]:=Y[1]://para o gráfico
x2[0]:=X[2]:y2[0]:=Y[2]:
x3[0]:=X[3]:y3[0]:=Y[3]:

g:=1://para coord. reduzidas, senão g = 6.67.10^(-11)

h:=dt/2://passo inicial

for r from 0 to passos do// número de pontos a traçar
  for i from 1 to n-1 do//velocidades para h:=dt/2
    for k from i+1 to n do

      c:=g*m[i]:

      xx:=x[i]-x[k]:
      yy:=y[i]-y[k]:
      r3:=(xx*xx+yy*yy)^(-3/2):
      c1:=c*m[k]*r3:
      fx:=-c1*xx:
      fy:=-c1*yy:
      u[i]:=u[i]+fx*h/m[i]:u[k]:=u[k]-fx*h/m[k]:
      v[i]:=v[i]+fy*h/m[i]:v[k]:=v[k]-fy*h/m[k]:

    end_for:
  end_for:
end_for:

```

```

h:=dt:
for j from 1 to n do// Coordenadas x e y
x[j]:=x[j]+u[j]*h:
y[j]:=y[j]+v[j]*h:
end_for:
//pontos para o gráfico
x1[r]:=x[1]:y1[r]:=y[1]:
x2[r]:=x[2]:y2[r]:=y[2]:
x3[r]:=x[3]:y3[r]:=y[3]:
x1[0]:=X[1]:y1[0]:=Y[1]:
  x2[0]:=X[2]:y2[0]:=Y[2]:
  x3[0]:=X[3]:y3[0]:=Y[3]:

end_for:

plot(plot::Group2d(
plot::Point2d(x1[r],y1[r],Color=RGB::Red)$ r=0..passos,
plot::Point2d(x2[r],y2[r],Color=RGB::Green)$ r=0..passos,
plot::Point2d(x3[r],y3[r],Color=RGB::Blue)$ r=0..passos,

  ViewingBox=[-1.5..2,-1..2],Scaling=Constrained )):
end_proc:

N_corpos(0.1,30)

```

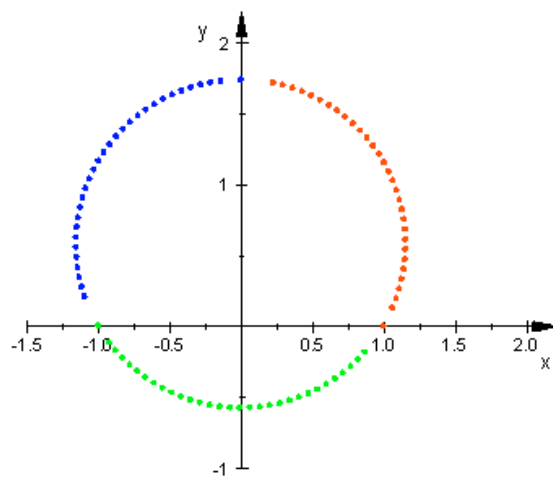


Fig.5.4-2

No capítulo 7, nos ocuparemos mais a fundo com o método de Feynman. Aqui o utilizamos porque não é difícil entendê-lo. O método `numeric::odesolve2` do MuPAD funciona muito bem, muito melhor do que o método de Feynman, mas não sabemos o que se move adentro. É como uma caixa eletrônica ou mágica: No "slit" para a inserção dos dados se coloca as equações junto com os valores iniciais, e, na porta da saída, aparecerão os resultados.

No Programa 3 metemos todos as informações do problema anterior, mas as equações são anotadas explicitamente. Esta lista extensa podemos facilmente produzir por meio do seguinte programa:

#### Programa 2:

- `reset()://acelerações no caso de n-corpos`
- ```
//r[i][j]:=sqrt((x[i]-x[j])^2+(y[i]-y[j])^2)//distâncias
ax[1]:=_plus(_plus(-m[j]*(x[i]-x[j])/r[i][j]^3 $i=1..1)$
j=1..4);
ay[1]:=_plus(_plus(-m[j]*(y[i]-y[j])/r[i][j]^3 $i=1..1)$
j=1..4);
//Para calcular ax[2],ay[2] etc. deve-se por "$i=2..2",
etc.
//Veja 4.7.2
ax[2]:=_plus(_plus(-m[j]*(x[i]-x[j])/r[i][j]^3 $i=2..2)$
j=1..4);
ay[2]:=_plus(_plus(-m[j]*(y[i]-y[j])/r[i][j]^3 $i=2..2)$
j=1..4);
```

Não podemos utilizar simplesmente o programa das duas bolas de tênis de 2.5.1, pois eles moveram-se sem interação mútua, e por isso pudemos escrever 2 IVP independentes, (IVP = Initial Value Problem problema de condições iniciais). Agora temos três corpos com interação e os três representam um só problema. Primeiro consideramos os três corpos do exemplo anterior.

## 5.4.2 Trajetórias de três corpos com MuPAD

### Programa 3:

```

• reset()://3 corpos
  m1:=1:m2:=1:m3:=1:
  DIGITS=20:

  IVP:={x1''(t)=
    -m2*(x1(t)-x2(t))/sqrt((x1(t)-x2(t))^2+(y1(t)-y2(t))^2)^3
    -m3*(x1(t)-x3(t))/sqrt((x1(t)-x3(t))^2+(y1(t)-y3(t))^2)^3,
  y1''(t)=
    -m2*(y1(t)-y2(t))/sqrt((x1(t)-x2(t))^2+(y1(t)-y2(t))^2)^3
    -m3*(y1(t)-y3(t))/sqrt((x1(t)-x3(t))^2+(y1(t)-y3(t))^2)^3,
  x2''(t)=
    -m1*(x2(t)-x1(t))/sqrt((x2(t)-x1(t))^2+(y2(t)-y1(t))^2)^3
    -m3*(x2(t)-x3(t))/sqrt((x2(t)-x3(t))^2+(y2(t)-y3(t))^2)^3,
  y2''(t)=
    -m1*(y2(t)-y1(t))/sqrt((x2(t)-x1(t))^2+(y2(t)-y1(t))^2)^3
    -m3*(y2(t)-y3(t))/sqrt((x2(t)-x3(t))^2+(y2(t)-y3(t))^2)^3,
  x3''(t)=
    -m1*(x3(t)-x1(t))/sqrt((x3(t)-x1(t))^2+(y3(t)-y1(t))^2)^3
    -m2*(x3(t)-x2(t))/sqrt((x3(t)-x2(t))^2+(y3(t)-y2(t))^2)^3,
  y3''(t)=
    -m1*(y3(t)-y1(t))/sqrt((x3(t)-x1(t))^2+(y3(t)-y1(t))^2)^3
    -m2*(y3(t)-y2(t))/sqrt((x3(t)-x2(t))^2+(y3(t)-y2(t))^2)^3,
  x1(0)=1,x2(0)=-1, x3(0)=0,//os 3 corpos do Programa1
  y1(0)=0,y2(0)=0,y3(0)=1.732,
  x1'(0)=0.35355,x2'(0)=0.35355,x3'(0)=-0.7071,
  y1'(0)=0.61237,y2'(0)=-0.61237,y3'(0)=0}:

/*x1(0)=-m2,x2(0)=m1, x3(0)=0, //Dados de MuPAD
  y1(0)=0.7*m3,y2(0)=0,y3(0)=-0.7*m1,
  x1'(0)=-1.01*m3,x2'(0)=0,x3'(0)=1.01*m1,
  y1'(0)=-0.9*m2,y2'(0)=0.9*m1,y3'(0)=0}:/

```



```

fields:=[x1(t),y1(t),x2(t),y2(t),x3(t),y3(t),
x1'(t),y1'(t),x2'(t),y2'(t),x3'(t),y3'(t)]:

ivp:=numeric::ode2vectorfield(IVP,fields):
Y:=numeric::odesolve2(ivp):
//Animation

dt:=0.05:imax:=60:

plot("//"Sol" para o exemplo de MuPAD
plot::Point2d(Y(t)[1], Y(t)[2],
Color = RGB::Blue,
VisibleFromTo = t..t + 0.99*dt,
PointSize = 2*unit::mm)
$ t in [i*dt $ i = 0..imax],
plot::Line2d([Y(t - dt)[1], Y(t - dt)[2]],
[Y(t)[1], Y(t)[2]], Color = RGB::Blue,
VisibleAfter = t)
$ t in [i*dt $ i = 1..imax],
//Planeta grande
plot::Point2d(Y(t)[3], Y(t)[4],
Color = RGB::Red,
VisibleFromTo = t..t + 0.99*dt,
PointSize = 2*unit::mm)
$ t in [i*dt $ i = 0..imax],
plot::Line2d([Y(t - dt)[3], Y(t - dt)[4]],
[Y(t)[3], Y(t)[4]], Color = RGB::Red,
VisibleAfter = t)
$ t in [i*dt $ i = 1..imax],
//Planeta pequeno
plot::Point2d(Y(t)[5], Y(t)[6],
Color = RGB::Green,
VisibleFromTo = t..t + 0.99*dt,
PointSize = 2*unit::mm)
$ t in [i*dt $ i = 0..imax],
plot::Line2d([Y(t - dt)[5], Y(t - dt)[6]],
[Y(t)[5], Y(t)[6]], Color = RGB::Green,
VisibleAfter = t)
$ t in [i*dt $ i = 1..imax],
Scaling=Constrained):

```

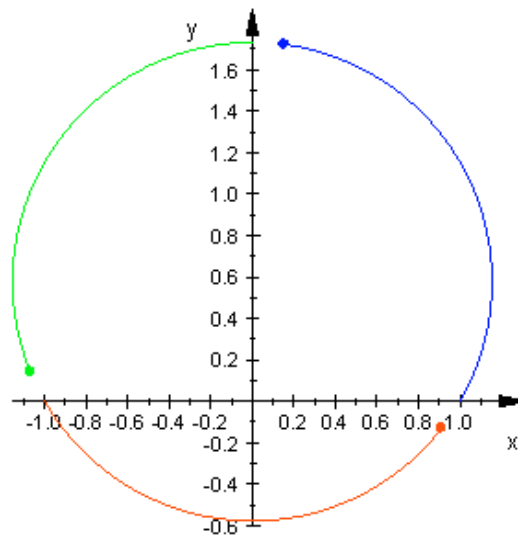


Fig.: 5.4-3

### 5.4.3 Trajetórias de 4 corpos

Abordemos agora o seguinte caso de 4 corpos.

Uma sonda espacial ( $m = 1000$  kg) é lançado no espaço a partir da Terra. A sonda partirá em direção a Vênus e depois de quatro meses, ela vai atingir a aproximação mínima do planeta Vênus. Os dados encontram-se na seguinte tabela:

	Massa/kg	Posição inicial/m	Velocidade inicial/ $\text{ms}^{-1}$
Sol	2E30	0,0	0,0
Vênus	4.85E24	9.553E10, -5.08E10	16440, 30920
Terra	5.98E24	1.496E11,0	0, 0.29770
Sonda	1000	1.49E11, 1E10	-5000, 27200

O programa a seguir é uma pequena variação do programa anterior, somente apresento as linhas principais. Os cálculos das interações podem ser feitos por meio do Programa 2.

### Programa 4:

- ```

reset()://4Corpos
m1:=2E30*g:m2:=4.85E24*g:m3:=5.98E24*g:m4:=1E3*g:
g:=6.67E-11:
DIGITS=20:

IVP:={x1''(t)=
-m2*(x1(t)-x2(t))/sqrt((x1(t)-x2(t))^2+(y1(t)-y2(t))^2)^3
-m3*(x1(t)-x3(t))/sqrt((x1(t)-x3(t))^2+(y1(t)-y3(t))^2)^3
-m4*(x1(t)-x4(t))/sqrt((x1(t)-x4(t))^2+(y1(t)-y4(t))^2)^3,
y1''(t)=
-m2*(y1(t)-y2(t))/sqrt((x1(t)-x2(t))^2+(y1(t)-y2(t))^2)^3
-m3*(y1(t)-y3(t))/sqrt((x1(t)-x3(t))^2+(y1(t)-y3(t))^2)^3
-m4*(y1(t)-y4(t))/sqrt((x1(t)-x4(t))^2+(y1(t)-y4(t))^2)^3,
x2''(t)=
-m1*(x2(t)-x1(t))/sqrt((x2(t)-x1(t))^2+(y2(t)-y1(t))^2)^3
-m3*(x2(t)-x3(t))/sqrt((x2(t)-x3(t))^2+(y2(t)-y3(t))^2)^3
-m4*(x2(t)-x4(t))/sqrt((x2(t)-x4(t))^2+(y2(t)-y4(t))^2)^3,
...
x3''(t)=
-m1*(x3(t)-x1(t))/sqrt((x3(t)-x1(t))^2+(y3(t)-y1(t))^2)^3
-m2*(x3(t)-x2(t))/sqrt((x3(t)-x2(t))^2+(y3(t)-y2(t))^2)^3
-m4*(x3(t)-x4(t))/sqrt((x3(t)-x4(t))^2+(y3(t)-y4(t))^2)^3,
...

```

```

x4''(t) =
-m1*(x4(t)-x1(t))/sqrt((x4(t)-x1(t))^2+(y4(t)-y1(t))^2)^3
-m2*(x4(t)-x2(t))/sqrt((x4(t)-x2(t))^2+(y4(t)-y2(t))^2)^3
-m3*(x4(t)-x3(t))/sqrt((x4(t)-x3(t))^2+(y4(t)-y3(t))^2)^3,
...

x1(0)=0,x2(0)=9.553E10, x3(0)=1.496E11,x4(0)=1.49E11,
y1(0)=0,y2(0)=-5.08E10,y3(0)=0,y4(0)=1E10,
x1'(0)=0,x2'(0)=16440,x3'(0)=0,x4'(0)=-5E3,
y1'(0)=0,y2'(0)=30920,y3'(0)=29770,y4'(0)=27200}:

fields:=[x1(t),y1(t),x2(t),y2(t),x3(t),y3(t),x4(t),y4(t),
x1'(t),y1'(t),x2'(t),y2'(t),x3'(t),y3'(t),x4'(t),y4'(t)]:

ivp:=numeric::ode2vectorfield(IVP,fields):
Y:=numeric::odesolve2(ivp):

```

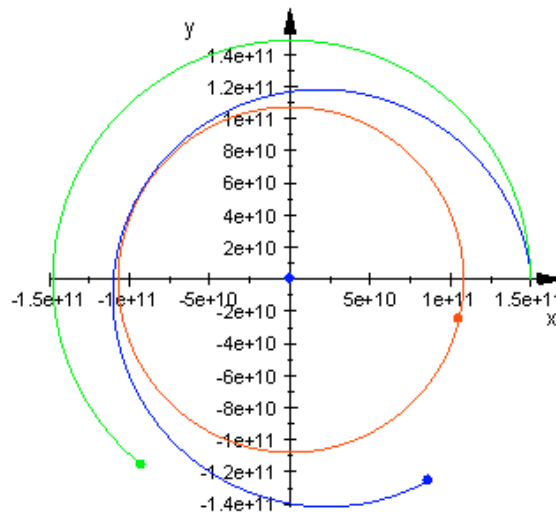


Fig.: 5.4-4

O Sol fica na origem e a Terra move-se na trajetória exterior (verde). A sonda tem a trajetória azul. O dt foi de 200 000 e imax = 100.

## 5.4.4 Trajetórias de três corpos com MuPAD (Exercícios)

Você pode, usando os programas anteriores, imaginar e analisar muitos pequenos "sistemas solares". O mais simple procedimento para isso seria variar os dados iniciais para os corpos em interação.

A seguinte figura mostra que as órbitas dos planetas Terra e Vênus são praticamente círculos com o Sol no entro.

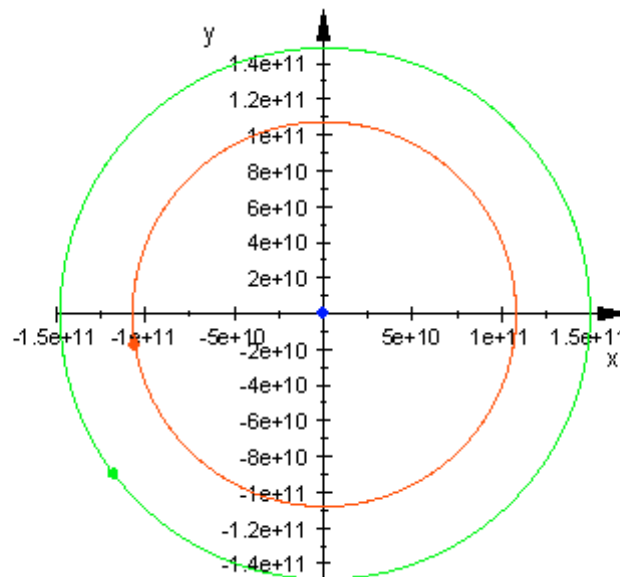


Fig.: 5.4-5

```
m1:=2E30*g:m2:=4.85E24*g:m3:=5.98E24*g:
```

```
g:=6.67E-11:DIGITS=30:
```

```
x1(0)=0,x2(0)=9.553E10,x3(0)=1.496E11,
```

```
y1(0)=0,y2(0)=-5.08E10,y3(0)=0,
```

```
x1'(0)=0,x2'(0)=16440,x3'(0)=0,
```

```
y1'(0)=0,y2'(0)=30920,y3'(0)=29770}:
```

```
dt:=2000000; imax := 100 ou mais
```

No Manual de MuPAD, você pode encontrar o exemplo de dois massas  $m_1$ ,  $m_2$  que se movem ao redor de um "Sol" de massa muito grande. As condições iniciais são

```

m1:=1:m2:=0.04:m3:=0.0001: (m1:= "Sol")
x1(0)=-m2,x2(0)=m1,x3(0)=0,
y1(0)=0.7*m3,y2(0)=0,y3(0)=-0.7*m1,
x1'(0)=-1.01*m3,x2'(0)=0,x3'(0)=1.01*m1,
y1'(0)=-0.9*m2,y2'(0)=0.9*m1,y3'(0)=0}:

```

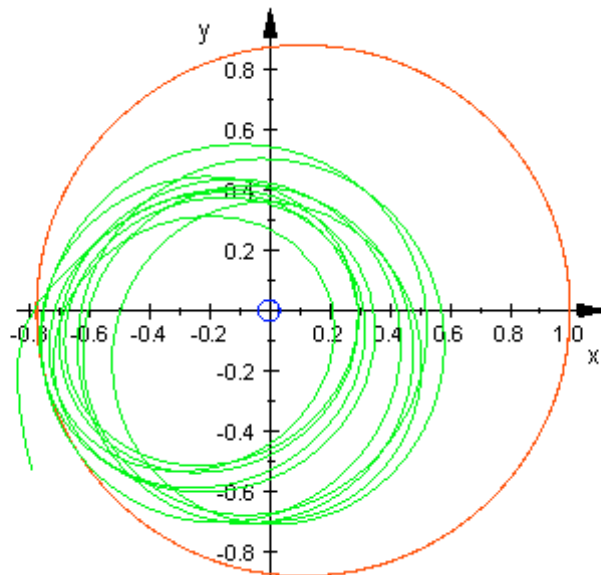


Fig.: 5.4-6

A figura mostra que o planeta pequeno com  $m_3 := 0.0001$  vai sair do sistema dos três corpos. A trajetória do maior corpo de massa  $m_1 := 1$  sofre apenas pequenas perturbações.

O seguinte **Programa 5** calcula outra vez a trajetória do planeta Mercúrio cuja órbita já foi considerada no parágrafo 5.2.1, utilizando o `numeric::odesolve2` do MuPAD. Esta vez estamos baseando-nos só nas definições da velocidade e da aceleração. Tente de compreender o programa usando as relações

$$v(t) \approx (x(t+h) - x(t))/h \quad e \quad a(t) \approx (v(t+h) - v(t))/h, \text{ onde } h = \Delta t, \text{ veja 5.4.1}$$

Quanto menor é o intervalo de tempo  $h$ , tanto melhor aproximam estas expressões as derivadas exatas  $v(t)$  e  $a(t)$ . O algoritmo é chamado segundo Euler.

- `reset()` :  
`Euler:=proc(h,passos)//Mercúrio /confira 5.2.2`  
`begin`  
`t(0):=0:i:=0:`  
`x(0):=0.3075:y(0):=0:u(0):=0:v(0):=1.982:`  
`DIGITS:=6:`  
`print("t= ",t(i),"x= ",x(i),"y= ",y(i)):`  
  
`for i from 1 to passos do`  
`r(i):=sqrt(x(i-1)^2+y(i-1)^2):`  
`ax:=-x(i-1)/r(i)^3:`  
`ay:=-y(i-1)/r(i)^3:`  
`u(i):=u(i-1)+ax*h:`  
`v(i):=v(i-1)+ay*h:`  
`x(i):=x(i-1)+u(i)*h:`  
`y(i):=y(i-1)+v(i)*h:`  
`t(i):=t(i-1)+h:`  
`print("t= ",t(i),"x= ",x(i),"y= ",y(i)):`  
  
`end_for:`  
  
`//plot(plot::Point2d([x(i),y(i)])$`  
`i=0..passos,Color=RGB::Blue):`  
`end_proc:`  
`Euler(0.01,5)`

**Resultados:**

```

"t= ", 0, "x= ", 0.3075, "y= ", 0
"t= ", 0.01, "x= ", 0.306442, "y= ", 0.01982
"t= ", 0.02, "x= ", 0.304327, "y= ", 0.0395716
"t= ", 0.03, "x= ", 0.301158, "y= ", 0.0591862
"t= ", 0.04, "x= ", 0.296947, "y= ", 0.0785961
"t= ", 0.05, "x= ", 0.291713, "y= ", 0.0977349

```

**5.4.5 Trajetória Lua-Terra (Problema restrito)**

No chamado "**Problema restrito** de três corpos" movem-se dois corpos pesados em torno do centro de massa comum enquanto um terceiro corpo leve move-se no mesmo plano que os corpos pesados. Podemos imaginar-nos uma sonda espacial  $m_3$  que se move no campo gravitacional da Terra  $m_1$  e da Lua  $m_2$ . A influencia do Sol não é tomada em conta.

Na figura 5.4-7 vemos Terra e Lua sobre o eixo-x de um sistema de coordenadas que gira com velocidade angular  $\omega$  constante.

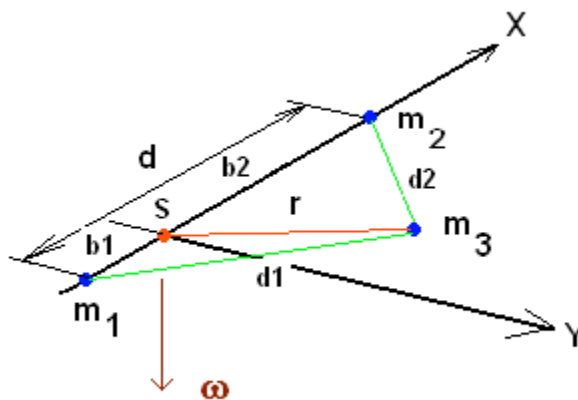


Fig.: 5.4-7



Os dois planetas descrevem círculos coplanares em torno do seu centro de massa. Com isso, temos reduzido um problema de 3 corpos a um de um corpo só e a nossa tarefa consiste em calcular posição e velocidade de  $m_3$  no transcurso do tempo.

A Terra tem do Sol a distância  $b_1 = m \cdot d$ , sendo  $m := m_2 / (m_1 + m_2)$ . A distância entre Sol e Lua é  $b_2 = m' \cdot d$  com  $m' = 1 - m$ . A velocidade angular tem a direção do eixo-z e o valor  $\omega^2 = G (m_1 + m_2) / d^3$ .

Num sistema inercial, a segunda Lei de Newton rezaria  $m_3 \mathbf{a} = \mathbf{F}_1 + \mathbf{F}_2$ , onde  $\mathbf{F}_1$  e  $\mathbf{F}_2$  são as forças devido a  $m_1$  e  $m_2$ . Em nosso sistema não inercial temos que introduzir duas forças "inerciais", porque o sistema não é inercial (!), confira 2.1.3. São a força centrífuga:  $\mathbf{F}_c = -m_3 \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r})$  e a força CORIOLIS  $\mathbf{F}_{cor} = -2 m_3 \boldsymbol{\omega} \times \mathbf{v}_{rel}$ .

As equações de movimento para as duas coordenadas são (compare com 2.5.3)

$$\frac{d^2 x}{dt^2} = x + 2 \frac{dy}{dt} - \frac{m'(x+m)}{d_1^3} - \frac{m(x-m')}{d_2^3} \quad (6)$$

$$\frac{d^2 y}{dt^2} = y - 2 \frac{dx}{dt} - \frac{m'y}{d_1^3} - \frac{my}{d_2^3} \quad (7)$$

A unidade do tempo foi escolhido de tal forma para que  $\omega = 1$ , ou seja para que o tempo para uma rotação do sistema de coordenadas fosse  $T = 2\pi$ .

As distâncias  $d_1$  e  $d_2$  são

$$d_1^2 = (b_1 + x)^2 + y^2 = (m \cdot d + x)^2 + y^2 \quad (8)$$

$$d_2^2 = (b_2 - x)^2 + y^2 = (m \cdot d - x)^2 + y^2 \quad (9)$$

Com  $d = 1$  a massa torna-se  $m = 0.012277471$ .

As condições iniciais serão  $x_0 = 0.994$ ,  $y_0 = 0$ .

$dx(0)/dt := vx(0) = 0$  e  $dy(0)/dt := vy(0) = -2.1138987966945$ .

Esta enorme quantidade de casas decimais é necessária, pois os cálculos são muito sensíveis com respeito a variações delas. Nos primeiros tempos dos voos espaciais foi absolutamente necessário de não permitir uma diferença da "injection speed" de 10840 m/s por mais de 1 m/s. Com uma diferença de  $> 2$  m/s, a Lua não houvesse podido ser atingido, pois não houve possibilidade para corrigir a trajetória durante o voo. Você pode estudar agora, usando o seguinte programa, estes fatos.

## Programa 6:

```

• x0:=0.994://problema restrito de 3 corpos
  y0:=0:vx0:=0: vy0:=-2.1138987966945027:
  m:=0.012277471: mu:=1-m:
  r1:=((x+m)^2+y*y)^(3/2):
  r2:=((x-mu)^2+y*y)^(3/2):

  IVP:={x'(t)=x(t)+2*y'(t)-mu*(x(t)+m)/r1-m*(x(t)-mu)/r2,
        y'(t)=y(t)-2*x'(t)-mu*y(t)/r1-m*y(t)/r2,
        x(0)=x0,x'(0)=vx0,
        y(0)=y0,y'(0)=vy0}:
  fields:=[x(t),y(t),x'(t),y'(t)]:

  ivp:=numeric::ode2vectorfield(IVP, fields):
  Y := numeric::odesolve2(ivp):

  //Animation
  dt:=0.005:imax:=990:

  plot(

  plot::Point2d(-0.01228,0, Color=RGB::Black,
  PointSize=3*unit::mm),//Terra
  plot::Point2d(0.9877,0, Color=RGB::Green,
  PointSize=2*unit::mm), //Lua

  plot::Point2d(Y(t)[1], Y(t)[2],
  Color = RGB::Blue,VisibleFromTo = t..t + 0.99*dt,
  PointSize = 2*unit::mm) $ t in [i*dt $ i = 0..imax],

  plot::Line2d([Y(t - dt)[1], Y(t - dt)[2]],
  [Y(t)[1], Y(t)[2]], Color = RGB::Red,VisibleAfter = t)
  $ t in [i*dt $ i = 1..imax]):

```

Para a figura 5.4.9 foram usados  $dt = 0.005$  e  $imax = 3000$ .

A velocidade  $vy(0)$  teve o valor  $-2.0325$ . A trajetória resultante é muito mais comprida que no caso da figura 5.4.-8, e não é fechada.

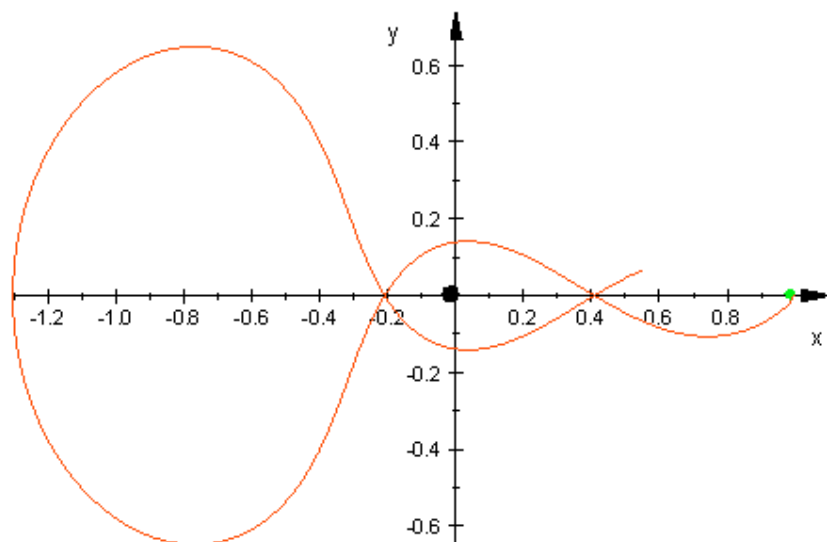


Fig.: 5.4-8

Posição da Terra:  $x(0) = -0.01228$ ,  $y(0) = 0$  e da Lua:  $x(0) = 0.9887$ ,  $y(0) = 0$ . A sonda parte em  $x(0) = 0.994$  e  $y(0) = 0$  do lado direito da Lua.

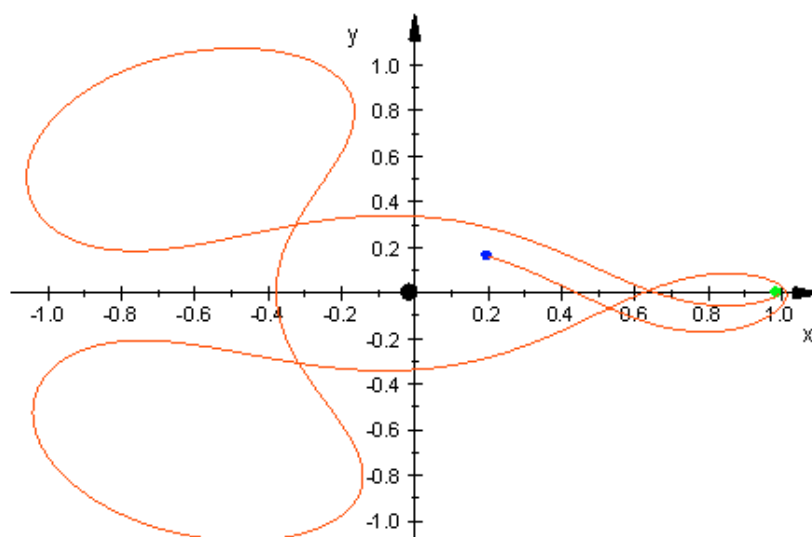


Fig.: 5.4-9