

Projeto 5 - LEIS DE KEPLER E O PROBLEMA DE TRÊS CORPOS

Instituto de Física de São Carlos
Universidade de São Paulo

Gabriel Lima Alves (12558547)

Introdução à Física Computacional
Prof. Francisco Castilho Alcaraz

Dezembro, 2022



Tarefa A

Tarefa A1

Na tarefa A é pedido para calcular as posições $(x(t), y(t))$ dos planetas do sistema solar em uma simulação de orbita circular, sendo que o sistema simulado é um sistema de dois corpos (sol e o determinado planeta). Essa simulação é realizada utilizando o método de Método de Verlet, que se baseia na expansão Taylor, que para a componente y pode ser difiida como:

$$y(t_i + \Delta t) = y(t_i) \pm \frac{dy}{dt_i} \Delta t + \frac{1}{2} \frac{d^2 y}{dt_i^2} (\Delta t)^2 \pm \frac{1}{6} \frac{d^3 y}{dt_i^3} (\Delta t)^3 \dots$$

A equação da posição y em da orbita em função do tempo definida a partir do Método de Verlet pode ser vista abaixo.

$$y_{i+1} = 2y_i - y_{i-1} + \frac{d^2 y}{dt_i^2} (\Delta t)^2 + O(\Delta t)^4$$

O objetivo da simulação era encontrar, utilizando o raio médio da orbita dos planetas do sistema solar, a velocidade que define a melhor orbita circular, ou seja, a orbita com menor excentricidade possível. Além disso, também foi encontrado o período dos planetas, bem como verificado a seguinte relação $\frac{T^2}{R^3}$.

O algoritmo utilizado para a encontrar a solução da orbita dos planetas pode ser visto a seguir.

```
1  c      program tarefa-A
2      implicit real*8 (a-h,o-z)
3      Parameter(nite = 91250d0) !numero de interações
4      Parameter(tempo = 250d0) !anos
5      Parameter(rpi = dacos(-1d0)) !pi
6      dimension x(0:2)
7      dimension y(0:2)
8      dimension r0(9)
9
10     Parameter(isaid1 = 10)
11     Parameter(isaid2 = 20)
12     Parameter(isaid3 = 30)
13     Parameter(isaid4 = 40)
14     Parameter(isaid5 = 50)
15     Parameter(isaid6 = 60)
16     Parameter(isaid7 = 70)
17     Parameter(isaid8 = 80)
18     Parameter(isaid9 = 90)
19     open(unit=isaid1,file='tarefa-1-saida-1-12558547.dat')
20     open(unit=isaid2,file='tarefa-1-saida-2-12558547.dat')
21     open(unit=isaid3,file='tarefa-1-saida-3-12558547.dat')
22     open(unit=isaid4,file='tarefa-1-saida-4-12558547.dat')
23     open(unit=isaid5,file='tarefa-1-saida-5-12558547.dat')
24     open(unit=isaid6,file='tarefa-1-saida-6-12558547.dat')
25     open(unit=isaid7,file='tarefa-1-saida-7-12558547.dat')
```

```

26     open(unit=isaid8,file='tarefa-1-saida-8-12558547.dat')
27     open(unit=isaid9,file='tarefa-1-saida-9-12558547.dat')
28
29     r0 = (/0.39d0,0.72d0,1d0,1.52d0,5.20d0,9.24d0,
30 +19.19d0,30.06d0,39.53d0/)
31
32     do j = 1,9
33
34         x(0) = r0(j)
35         y(0) = 0d0
36
37         v0x = 0d0
38         v0y = sqrt((4d0*(rpi**2))/(r0(j)))
39
40         dt = (tempo*1d0)/(nite*1d0)
41
42         !metodo de euler
43         x(1) = x(0) + v0x*dt
44         y(1) = y(0) + v0y*dt
45
46         write(10*j,*) x(0),y(0)
47         write(10*j,*) x(1),y(1)
48
49         icon = 0
50         rsum = 0d0
51         tsum = 0
52         taux = 0
53
54         do i = 2,nite
55
56             r = sqrt(x(1)**2+y(1)**2)
57             Gms = ((4d0*(rpi**2))/(r**3))
58
59             y(2) = 2d0*y(1) - y(0) -Gms*y(1)*(dt**2)
60             x(2) = 2d0*x(1) - x(0) -Gms*x(1)*(dt**2)
61
62             write(10*j,*) x(2),y(2)
63
64             if (y(1)*y(2) < 0) then
65                 icon = icon + 1
66
67                 rsum = rsum + abs(r)
68
69                 tsum = tsum + (dt*i - taux)
70
71                 taux = dt*i
72             end if
73

```

```

74         x(0) = x(1)
75         y(0) = y(1)
76         x(1) = x(2)
77         y(1) = y(2)
78
79     end do
80
81     a = rsum/(1d0*icont)
82
83     c = abs(a - r0(j))
84
85     t = 2d0*(tsum/(1d0*icont))
86
87     write(*,*) "P riodo: ", t
88
89     write(*,*) "Excentricidade: ", c/a
90
91     write(*,*) "P riodo em dias: ", t*365
92
93     write(*,*) "T2/r3: ", ((t**2)/(r0(j)**3))
94
95     write(*,*) "-----"
96
97 end do
98
99 close(isaid1)
100 close(isaid2)
101 close(isaid3)
102 close(isaid4)
103 close(isaid5)
104 close(isaid6)
105 close(isaid7)
106 close(isaid8)
107 close(isaid9)
108
109 end program

```

Algoritmo 1: C digo para resolu  o da tarefa A

Resultados

A seguir est o os gr ficos com as  rbitas dos planetas e os resultados descobertos pelo algoritmo.

Orbita de Mercúrio, Vênus e Terra

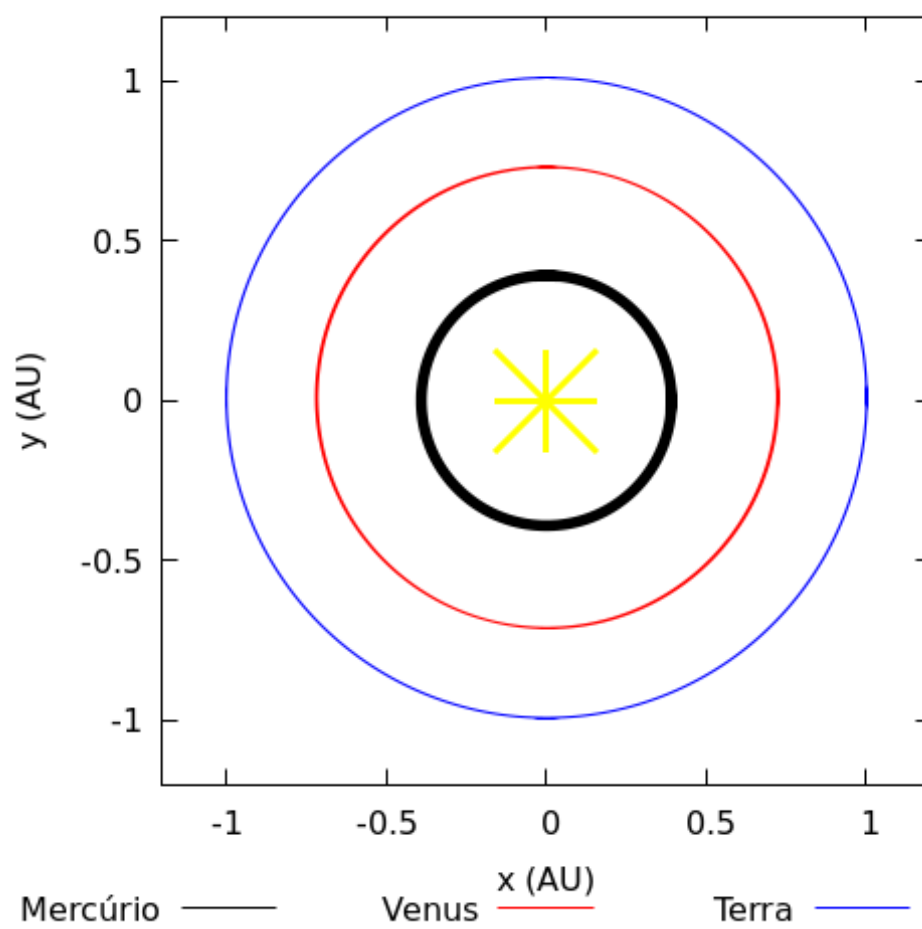


Figura 1: Orbita de Mercúrio, Vênus e Terra em torno do sol calculados isoladamente

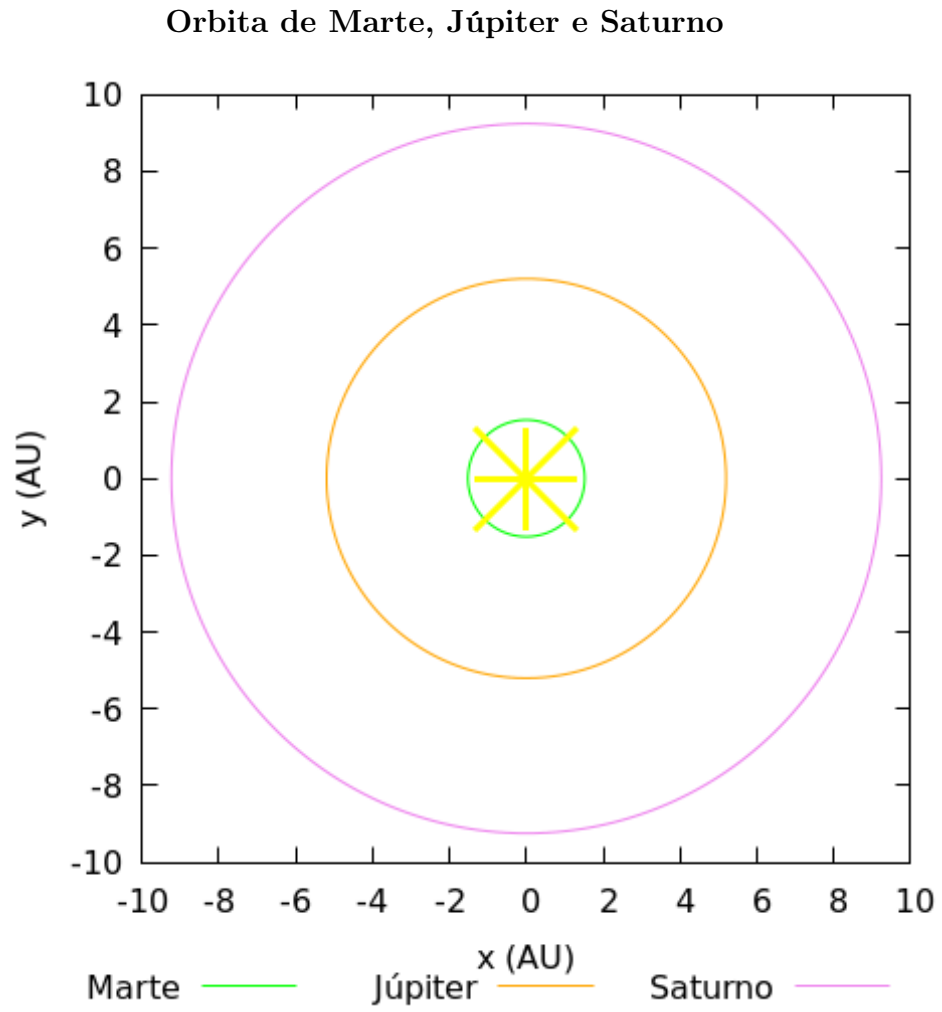


Figura 2: Orbita de Marte, Júpiter e Saturno em torno do sol calculados isoladamente

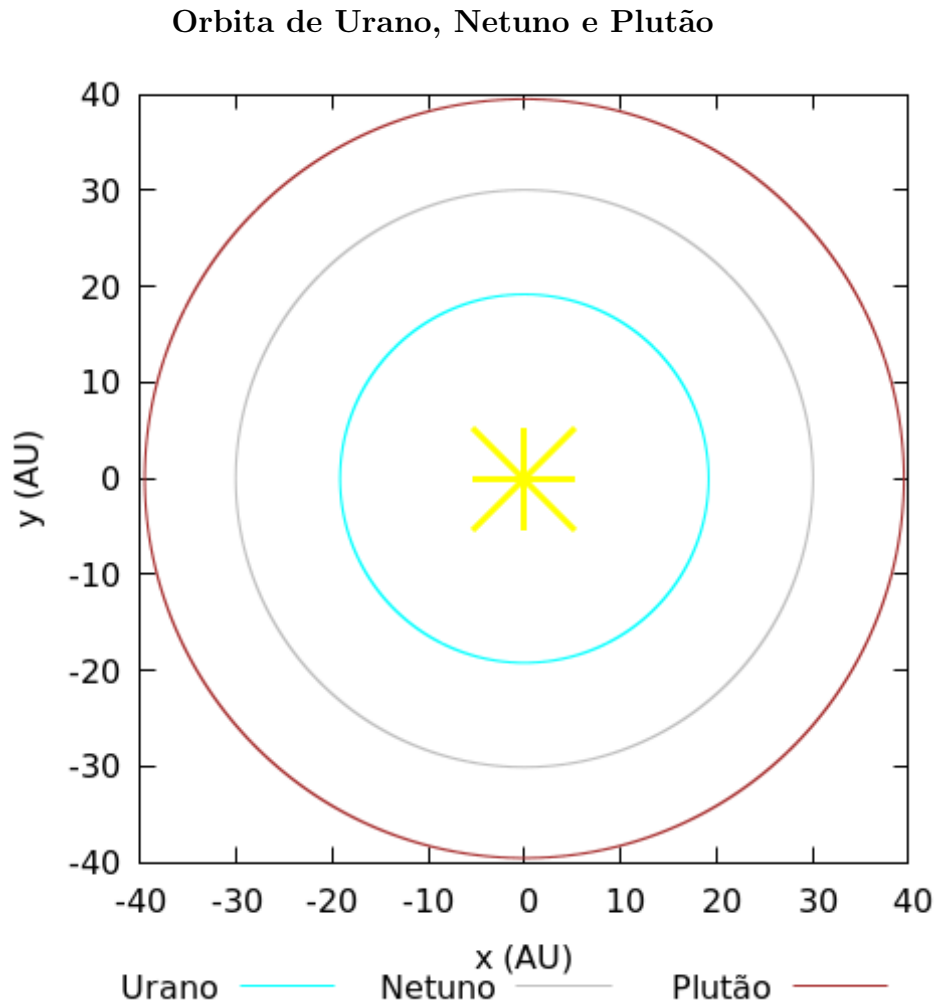


Figura 3: Orbita de Urano, Netuno e Plutão em torno do sol calculados isoladamente

Os valores de Δt usados para encontrar os resultados são múltiplos de um ano terrestre e o número de iterações múltiplos da quantidade de dias de um ano terrestre.

Logo em seguida, estão respectivamente os resultados dos seguintes planetas Mercúrio, Vênus, Terra, Marte, Júpiter, Saturno, Urano, Netuno e Plutão.

```

Périodo: 0.24442107378504202
Excentricidade: 1.8720274024175594E-003
Périodo em dias: 89.213691931540339
 $T^2/r^3$  1.0071252264912247
-----
Périodo: 0.61129759729045452
Excentricidade: 2.1121698450108381E-004
Périodo em dias: 223.12362301101589
 $T^2/r^3$  1.0011701401027808
-----
Périodo: 1.0002360886155872
Excentricidade: 7.3997063794316472E-005
Périodo em dias: 365.08617234468932
 $T^2/r^3$  1.0004722329690088
-----
Périodo: 1.8740961994026162
Excentricidade: 2.1256639754968952E-005
Périodo em dias: 684.04511278195491
 $T^2/r^3$  1.0001220353206468
-----
Périodo: 11.857925636007828
Excentricidade: 5.4804250565777297E-007
Périodo em dias: 4328.1428571428569
 $T^2/r^3$  1.0000170714972947

```

Figura 4: saída do algoritmo parte 1


```

Périodo:      28.087993553585818
Excentricidade: 1.0148069406092847E-007
Périodo em dias: 10252.117647058823
T²/r³      1.0000587634824014
-----
Périodo:      84.066849315068495
Excentricidade: 1.7984517795406670E-008
Périodo em dias: 30684.400000000001
T²/r³      1.0000566865913001
-----
Périodo:      164.81461187214612
Excentricidade: 4.8999926224141306E-009
Périodo em dias: 60157.333333333336
T²/r³      1.0000564042514914
-----
Périodo:      248.53698630136986
Excentricidade: 9.3801940576680880E-010
Périodo em dias: 90716.000000000000
T²/r³      1.0000037134736346
-----

```

Figura 5: saída do algoritmo parte 2

Analisando os resultados é possível perceber que os resultados foram próximo do esperado, os períodos por exemplo são equivalentes aos tabelados para cada planeta.

Tarefa B

Tarefa B1 e B2

O objetivo da tarefa B1 é similar à tarefa A1 porém agora a simulação é para três corpos (Sol, Terra e Júpiter). Ainda será utilizado o Método de Verlet.

Para a resolução da tarefa B2 foi utilizado o mesmo código da tarefa B1, apenas foi alterado a massa do planeta Júpiter e as devidas alterações na relação entre a massa do sol e Júpiter.

O código com as implementações descritas pode ser visto a seguir.

```

1  c      program tarefa-B1-B2
2          implicit real*8 (a-h,o-z)
3          Parameter(nite = 1000d0) !numero de interações
4          Parameter(tempo = 50d0) !anos
5          Parameter(rpi = dacos(-1d0)) !pi
6          dimension xt(0:2)
7          dimension yt(0:2)

```

```

8      dimension xj(0:2)
9      dimension yj(0:2)
10
11     Parameter(isaid1 = 10)
12     open(unit=isaid1,file='tarefa-3-saida-1-12558547.dat')
13
14     r0t = 1d0
15     r0j = 5.20d0
16
17     xt(0) = r0t
18     yt(0) = 0d0
19
20     xj(0) = r0j
21     yj(0) = 0d0
22
23     v0xt = 0d0
24     v0yt = sqrt((4d0*(rpi**2))/(r0t))
25
26     v0xj = 0d0
27     v0yj = sqrt((4d0*(rpi**2))/(r0j))
28
29     dt = (tempo*1d0)/(nite*1d0)
30
31     !metodo de euler terra
32     xt(1) = xt(0) + v0xt*dt
33     yt(1) = yt(0) + v0yt*dt
34
35     !metodo de euler jupiter
36     xj(1) = xj(0) + v0xj*dt
37     yj(1) = yj(0) + v0yj*dt
38
39     write(isaid1,*) xt(0),yt(0),xj(0),yj(0)
40     write(isaid1,*) xt(1),yt(1),xj(1),yj(1)
41
42     do i = 2,nite
43         !raios
44         rjs = sqrt(xj(1)**2+yj(1)**2)
45         rts = sqrt(xt(1)**2+yt(1)**2)
46         rtj = sqrt((xt(1)-xj(1))**2+(yt(1)-yj(1))**2)
47
48         !terra
49         Gms_t = ((4d0*(rpi**2))/(rts**3))
50         Gmj_t = (((4d0*(rpi**2))/(10**3))/(rtj**3))
51
52
53         yt(2) = 2d0*yt(1) - yt(0)
54         ++(-Gms_t*yt(1) - Gmj_t*(yt(1)-yj(1)))*(dt**2)
55         xt(2) = 2d0*xt(1) - xt(0)

```

```

56      ++(-Gms_t*xt(1) - Gmj_t*(xt(1)-xj(1)))*(dt**2)
57
58      !jupiter
59      Gms_j = ((4d0*(rpi**2))/(rjs**3))
60      Gmt_j = (((4d0*(rpi**2))/(3*10**5))/(rtj**3))
61
62      yj(2) = 2d0*yj(1) - yj(0)
63      ++(-Gms_j*yj(1) - Gmt_j*(yj(1)-yt(1)))*(dt**2)
64      xj(2) = 2d0*xj(1) - xj(0)
65      ++(-Gms_j*xj(1) - Gmt_j*(xj(1)-xt(1)))*(dt**2)
66
67      write(isaid1,*) xt(2),yt(2),xj(2),yj(2)
68
69      !terra
70      xt(0) = xt(1)
71      yt(0) = yt(1)
72      xt(1) = xt(2)
73      yt(1) = yt(2)
74
75      !jupiter
76      xj(0) = xj(1)
77      yj(0) = yj(1)
78      xj(1) = xj(2)
79      yj(1) = yj(2)
80
81      end do
82
83      close(isaid1)
84
85      end program

```

Algoritmo 2: Código para resolução da tarefa B1 e B2

Resultados

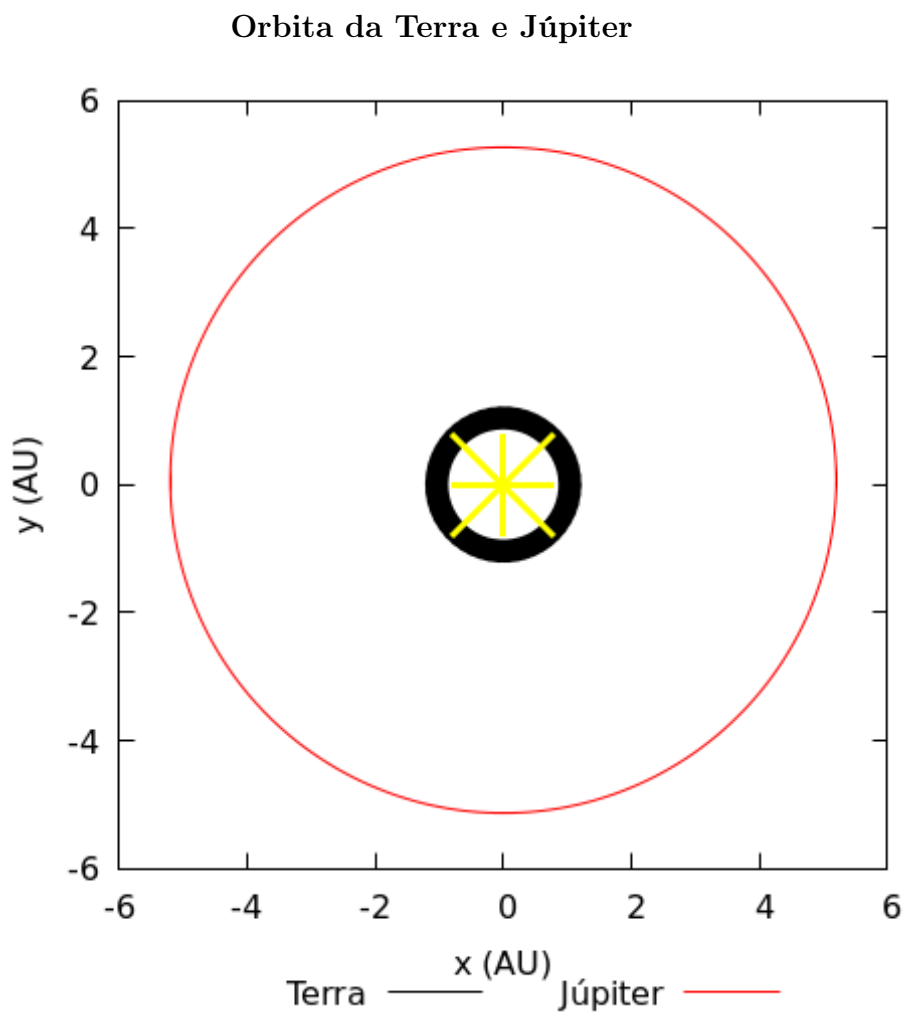


Figura 6: Resolução B1: Terra e Júpiter se influenciando e sendo influenciados pelo Sol

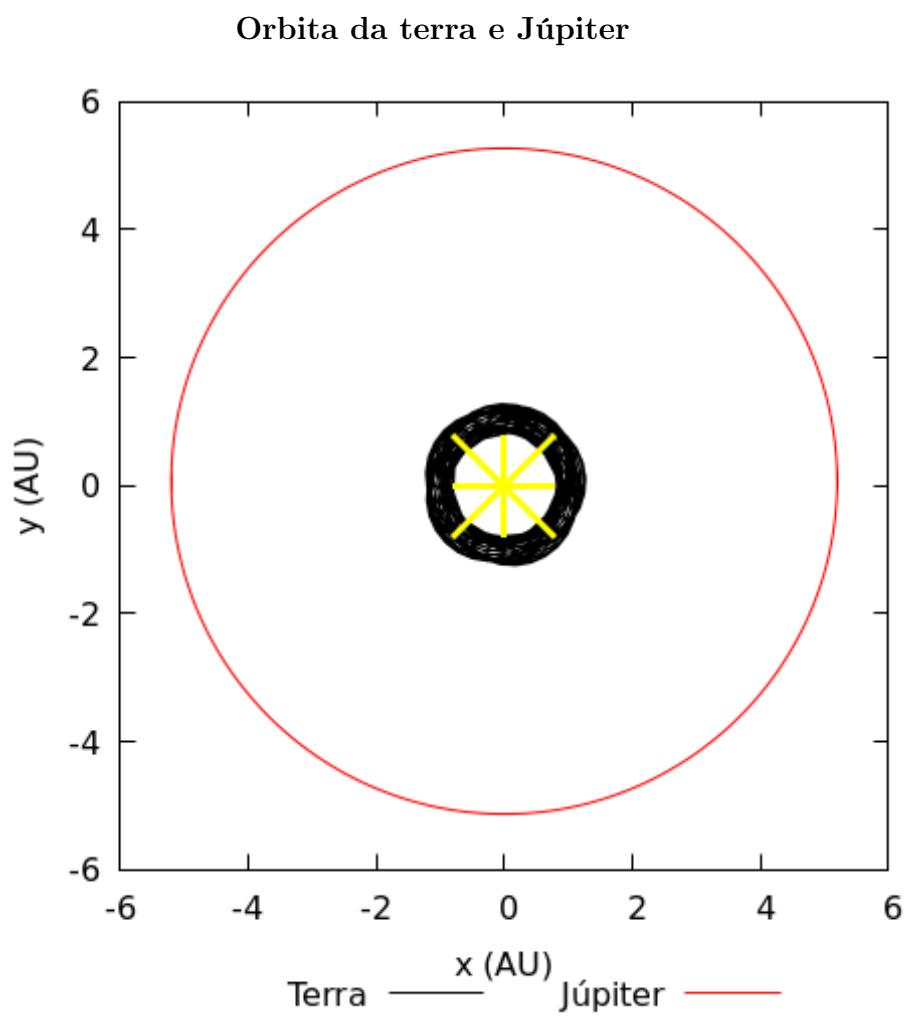


Figura 7: Resolução B2 (100x a massa de Júpiter): Terra e Júpiter se influenciando e sendo influenciados pelo Sol

Orbita da terra e Júpiter

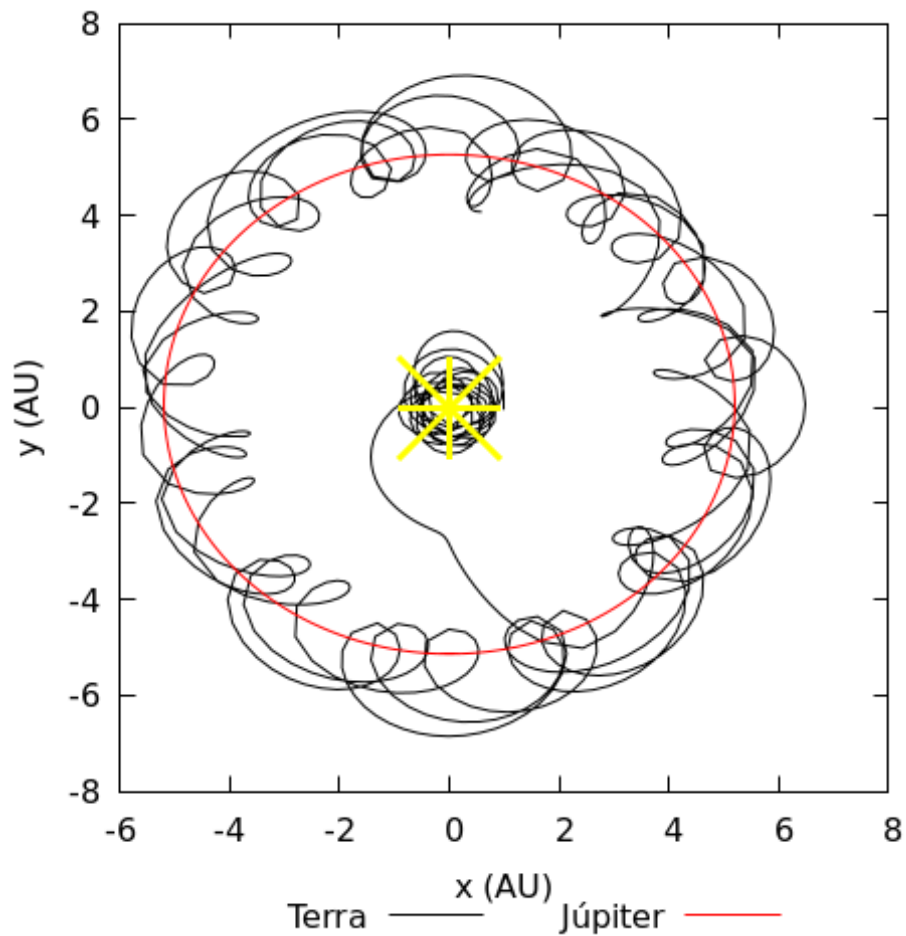


Figura 8: Resolução B2 (1000x a massa de Júpiter): Terra e Júpiter se influenciando e sendo influenciados pelo Sol

No gráfico resposta da tarefa B1 é possível ver que a orita da Terra não é uma linha como antes mas sim uma região de orbita isso acontece pois a orbita da Terra não é mais periódica e varia dentro da região por causa da influencia de força gravitacional de Júpiter.

O primeiro gráfico da tarefa B2 a orita da Terra continua não periódica e se torna mais instável devido ao aumento da massa do planeta de Júpiter e por consequência sua força gravitacional. No último gráfico da tarefa B2 é feito novamente um aumento da massa do planeta de Júpiter isso faz com que a orita da Terra fique ainda mais instável ao ponto dela colapsar.

Tarefa B3

A tarefa B3 é muito similar as tarefas B1 e B2, a diferença é que a simulação é feita com asteroides no lugar do planeta Terra. Dessa forma, foi calculado a orbita dos três asteroides supondo que apenas Júpiter e o sol tem efeito nos asteroides. Além disso, só foi considerado efeito do sol ao calcular a orbita de Júpiter.

O algoritmo com a solução pode ser visto abaixo.

```

1  c      program tarefa-B3
2          implicit real*8 (a-h,o-z)
3          Parameter(nite = 1000d0) !numero de interações
4          Parameter(tempo = 50d0) !anos
5          Parameter(rpi = dacos(-1d0)) !pi
6          dimension xt(0:2)
7          dimension yt(0:2)
8          dimension xj(0:2)
9          dimension yj(0:2)
10         dimension r0(3)
11         dimension v0ya(3)
12
13         Parameter(isaid1 = 10)
14         Parameter(isaid2 = 20)
15         Parameter(isaid3 = 30)
16         open(unit=isaid1,file='tarefa-4-saida-1-12558547.dat')
17         open(unit=isaid2,file='tarefa-4-saida-2-12558547.dat')
18         open(unit=isaid3,file='tarefa-4-saida-3-12558547.dat')
19
20         r0 = (/3.000d0,3.276d0,3.700d0/)
21         v0ya = (/3.628d0,3.471d0,3.267d0/)
22
23         do j = 1,3
24
25             r0t = r0(j)
26             r0j = 5.20d0
27
28             xt(0) = r0t
29             yt(0) = 0d0
30
31             xj(0) = r0j
32             yj(0) = 0d0
33
34             v0xt = 0d0
35             v0yt = v0ya(j)
36
37             v0xj = 0d0
38             v0yj = 2.755d0
39
40             dt = (tempo*1d0)/(nite*1d0)
41
42             !metodo de euler asteroides
43             xt(1) = xt(0) + v0xt*dt
44             yt(1) = yt(0) + v0yt*dt
45
46             !metodo de euler jupiter
47             xj(1) = xj(0) + v0xj*dt
48             yj(1) = yj(0) + v0yj*dt

```

```

49
50      write(10*j,*) xt(0),yt(0),xj(0),yj(0)
51      write(10*j,*) xt(1),yt(1),xj(1),yj(1)
52
53      do i = 2,nite
54          !raios
55          rjs = sqrt(xj(1)**2+yj(1)**2)
56          rts = sqrt(xt(1)**2+yt(1)**2)
57          rtj = sqrt((xt(1)-xj(1))**2+(yt(1)-yj(1))**2)
58
59          !asteroides
60          Gms_t = ((4d0*(rpi**2))/(rts**3))
61          Gmj_t = (((4d0*(rpi**2))/(10**3))/(rtj**3))
62
63
64          yt(2) = 2d0*yt(1) - yt(0)
65      ++(-Gms_t*yt(1) - Gmj_t*(yt(1)-yj(1)))*(dt**2)
66          xt(2) = 2d0*xt(1) - xt(0)
67      ++(-Gms_t*xt(1) - Gmj_t*(xt(1)-xj(1)))*(dt**2)
68
69          !jupiter
70          Gms_j = ((4d0*(rpi**2))/(rjs**3))
71          Gmt_j = (((4d0*(rpi**2))/(3*10**5))/(rtj**3))
72
73          yj(2) = 2d0*yj(1) - yj(0)
74      ++(-Gms_j*yj(1) - Gmt_j*(yj(1)-yt(1)))*(dt**2)
75          xj(2) = 2d0*xj(1) - xj(0)
76      ++(-Gms_j*xj(1) - Gmt_j*(xj(1)-xt(1)))*(dt**2)
77
78          write(10*j,*) xt(2),yt(2),xj(2),yj(2)
79
80          !asteroides
81          xt(0) = xt(1)
82          yt(0) = yt(1)
83          xt(1) = xt(2)
84          yt(1) = yt(2)
85
86          !jupiter
87          xj(0) = xj(1)
88          yj(0) = yj(1)
89          xj(1) = xj(2)
90          yj(1) = yj(2)
91
92      end do
93  end do
94
95  close(isaid1)
96  close(isaid2)

```



```

97         close(isaid3)
98
99     end program

```

Algoritmo 3: Código para resolução da tarefa B3

Resultados

A seguir está o gráfico com as orbitas de Júpiter e os três asteroides.

Orbita dos três asteroides e de Júpiter

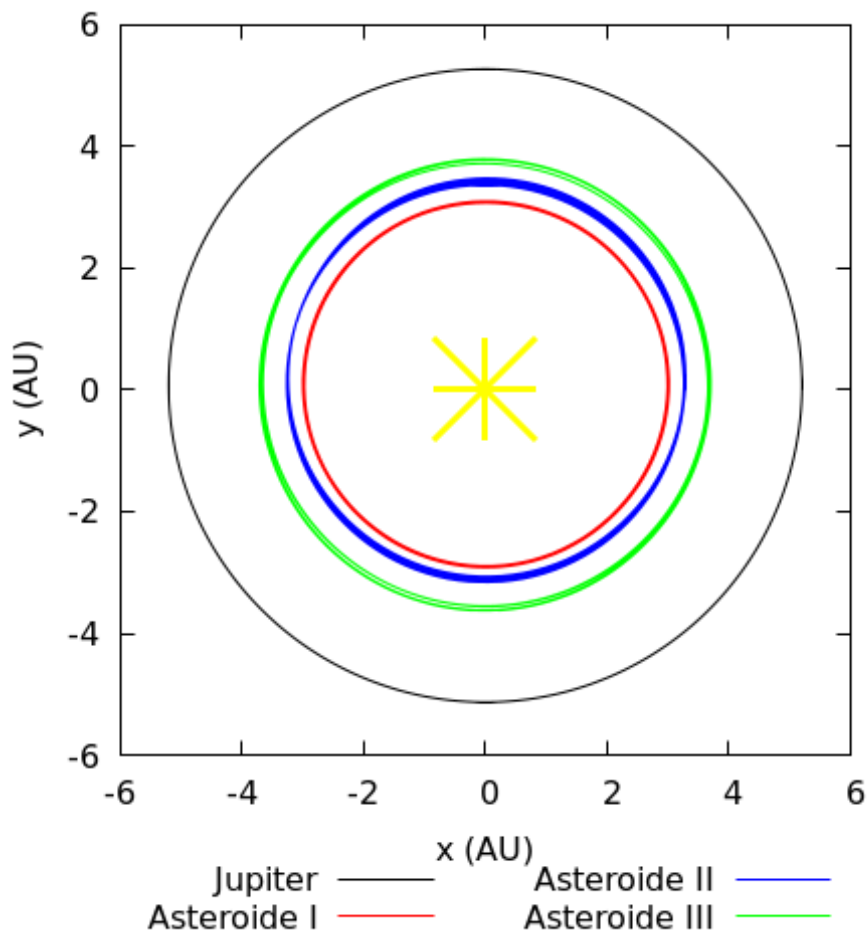


Figura 9: Orbita dos três asteroides e de Júpiter, considerando apenas a influencia de Júpiter e o Sol

Analisando o gráfico, é possível concluir que as orbitas dos asteroides são um pouco instáveis não formando uma linha de orbita bem definida.

Além disso, as lacunas de Kirkwood são espaços vazios no cinturão de asteroides, que correspondem a zonas de ressonância onde a atração gravitacional de Júpiter impede a permanência dos asteroides.