Celestial Mechanics – Solutions

Alexander V. Krivov & Tobias Stein[1]

Unit 5

Problem 5.1

In the circular restricted three-body problem the orbital velocity is given by

$$v^2 = n^2(x^2+y^2) + \underbrace{2\,G\left(\frac{M_1}{r_1}+\frac{M_2}{r_2}\right)}_{U} - C. \tag{1}$$

The two-body problem does *not* have a rotating reference frame. Hence, the first potential term, which contains the angular velocity n, vanishes; the Jacobi constant is replaced with the usual energy constant; and the secondary object vanishes:

$$\begin{aligned} n &\to 0, & (2)\\ C &\to -h = \frac{\kappa^2}{a} \approx \frac{GM_1}{a}, & (3)\\ M_2 &\to 0, & (4)\end{aligned}$$

from which the orbital velocity can be obtained:

$$v^2 = \kappa^2\left(\frac{2}{r}-\frac{1}{a}\right).$$

Letting $v = 0$, we can conclude that the zero-velocity surface is a sphere of radius

$$r = 2a.$$

Since $r_{\max} = r_{\mathrm{A}} = r(\theta = 180°) = a(1+e)$, the distance can be twice as large as the semimajor axis if and only if the orbit is a degenerate ellipse with $e = 1$ and the particle in its apocenter. That orbit corresponds to radial free fall with a start from rest.

Problem 5.2

As discussed in the lecture, in a special system of units ($a = 1$, $n = 1$) the expression for the Jacobi constant is

$$C = 2\left(\frac{1-\mu}{r_1}+\frac{\mu}{r_2}\right) + x^2 + y^2. \tag{5}$$

At the geometric center, it is obvious that $r_1 = r_2 = a/2$, and $y = 0$. Furthermore, we use units such that $a = 1$, so we get $r_1 = r_2 = 1/2$.
Inserting this into Eq. (5), we find

$$C = 4 + x^2.$$

We know that

$$x = -a_1 + a/2 = -a_1 + 1/2, \qquad a_1 + a_2 = a = 1, \qquad a_1M_1 = a_2M_2.$$

of which the latter can be written as $a_1(1-\mu) = a_2\mu$. This system of equations can be solved to give $x = -\mu + 1/2$ and finally

$$C = 4 + \left(\frac{1}{2}-\mu\right)^2.$$

[1]tobias.stein@uni-jena.de

Problem 5.3

See attached pages for a Wolfram Mathematica script that computes and plots Lagrangian points and zero-velocity surfaces.

Define distances, ZVS (+ derivatives), and Lagrangian points L1 to L7:

In[7]:=

```
r1[μ_, {x_, y_, z_}] := Sqrt[(x + μ)^2 + y^2 + z^2];
r2[μ_, {x_, y_, z_}] := Sqrt[(x - 1 + μ)^2 + y^2 + z^2];
ZVS[{μ_, β1_, n_}, {x_, y_, z_}] :=
  2 ((1 - μ) * (1 - β1) / r1[μ, {x, y, z}] + μ / r2[μ, {x, y, z}]) + n^2 * (x^2 + y^2);
(* derivatives: *)
ZVSx[{μ_, β_}, {x_, y_, z_}] := -(1 - μ) * (1 - β) * (x + μ) / r1[μ, {x, y, z}]^3 -
    μ * (x + μ - 1) / r2[μ, {x, y, z}]^3 + x;
ZVSy[{μ_, β_}, {x_, y_, z_}] :=
  y * (-(1 - μ) * (1 - β) / r1[μ, {x, y, z}]^3 + μ / r2[μ, {x, y, z}]^3 + 1);
ZVSz[{μ_, β_}, {x_, y_, z_}] :=
  z * (-(1 - μ) * (1 - β) / r1[μ, {x, y, z}]^3 - μ / r2[μ, {x, y, z}]^3);
LagrangianPoints[μ_, β1_] :=
  Module[{xz, xx, zz, μμ, rx, ry, rz, LC, β, aux = (1 - β1)^(1/3)},
   (* Find the general solutions for y=0 and z=0 *)
   rx = Solve[xx - (1 - μμ) * If[β1 == 0, 1, 1 - β] * (xx + μμ) / Abs[xx + μμ]^3 - μμ *
         (xx + μμ - 1) / Abs[xx + μμ - 1]^3 == 0 && (0 < μμ < 1) && β >= 0, xx, Reals];
   (* Drop the solutions that are invalid for different values of β1: *)
   rx = If[β1 > 1, Take[rx, {2}], If[β1 > 0, Drop[rx, {2, 3}], rx]];
   (* Insert the actual values for μ and β1: *)
   rx = xx /. (rx /. β → β1 /. μμ → μ);
   (* Generate y=0 values: *)
   ry = ConstantArray[0, Length[rx]];
   (* Add L4 and L5 if they exist: *)
   If[β1 < 1,
    (* β dependency according to Ammar (2008) *)
    AppendTo[rx, -μ + aux^2 / 2];
    AppendTo[ry, aux * Sqrt[1 - aux^2 / 4]];
    AppendTo[rx, rx // Last];
    AppendTo[ry, -(ry // Last)];
   ];
   (* Generate z=0 values: *)
   rz = ConstantArray[0, Length[rx]];
   (* Add co-planar points if they exist *)
   If[1 < β1 < 1 / (1 - μ),
    xz = {xx, zz} /. NSolve[ZVSx[{μ, β1}, {xx, 0, zz}] == 0 &&
         ZVSz[{μ, β1}, {xx, 0, zz}] == 0 && zz > 0, {xx, zz}, Reals] // Flatten;
    AppendTo[rx, xz[[1]]];
    AppendTo[rx, xz[[1]]];
    AppendTo[ry, 0];
    AppendTo[ry, 0];
    AppendTo[rz, xz[[2]]];
    AppendTo[rz, -xz[[2]]];
   ];
   (* Compute values of C at all points: *)
   LC = Map[ZVS[{μ, β1, 1}, #] &, Transpose[{rx, ry, rz}]];
```

```
  (* Sort by C: *)
  ReverseSort[Transpose[{rx, ry, rz, LC}], #2[[4]] > #1[[4]] &]
 ];
```

Compute {x, y, z, C} of the Lagrangian points (as many as exist; slow for 1<β1<1/(1-μ)):

In[13]:=
```
Manipulate[
 LagrangianPoints[μ, β1],
 {{μ, 0.2}, 0.01, 0.99}, {{β1, 0.0}, 0.0, 2.0}, SaveDefinitions → True
]
```

Out[13]=

μ

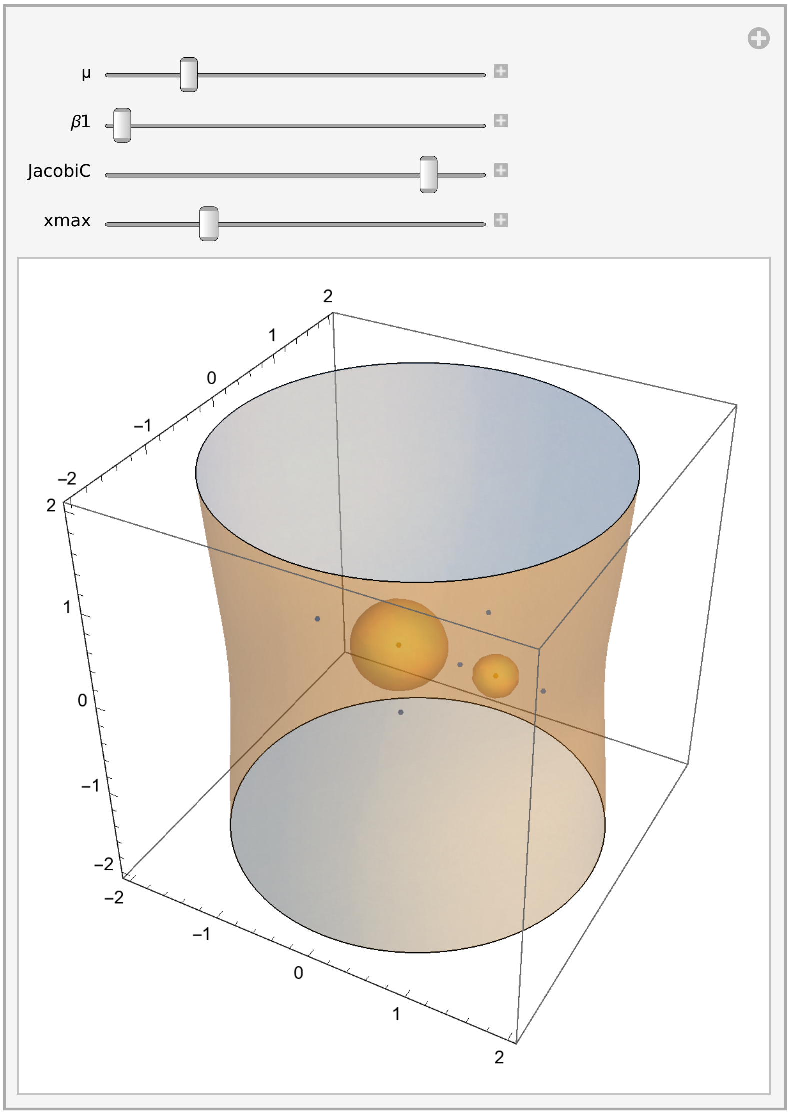
β1

```
{{0.438076, 0, 0, 3.80465},
 {1.27105, 0, 0, 3.55239}, {-1.08284, 0, 0, 3.19732},
 {0.3, 0.866025, 0, 2.84}, {0.3, -0.866025, 0, 2.84}}
```

Create rotatable 3D plots of the ZVS for different μ, C, and β1, with masses in red and Lagrangian points in blue:

In[14]:=
```
Manipulate[
 Show[
  RegionPlot3D[
   Evaluate[ZVS[{μ, β1, 1.0}, {x, y, z}] ≤ JacobiC],
   {x, -xmax, xmax}, {y, -xmax, xmax}, {z, -xmax, xmax},
   Mesh → None, PlotStyle → Directive[Opacity[0.3], Specularity[White, 2]],
   PlotPoints → 50
  ],
  ListPointPlot3D[
   {Map[Drop[#, -1] &, LagrangianPoints[μ, β1]], {{1 - μ, 0, 0}, {-μ, 0, 0}}}
  ]
 ],
 {{μ, 0.2}, 0.01, 0.99}, {{β1, 0.0}, 0.0, 2.0},
 {{JacobiC, 4.2}, -2.0, 5.0}, {{xmax, 2.0}, 1.0, 5.0}, SaveDefinitions → True
]
```

Out[14]=



Problem 5.4

Idk