CRTB Invariant Curves

Marc A. Murison

Astronomical Applications Department
U.S. Naval Observatory

murison@aa.usno.navy.mil
http://aa.usno.navy.mil/murison/

9 March, 1998

Introduction

restart

The Jacobi constant of the circular restricted three-body (CRTB) problem is

$$\Omega = \frac{\xi^2+\eta^2+\zeta^2}{2} + \frac{1-\mu}{r_1} + \frac{\mu}{r_2}$$

$JC := \%$

where $\mu = \frac{m_2}{m_1+m_2}$, $r_1 := \sqrt{(\xi+\mu)^2+\eta^2+\zeta^2}$, and $r_2 := \sqrt{(\xi-(1-\mu))^2+\eta^2+\zeta^2}$. The coordinates (ξ, η, ζ) are in the rotating frame. (Historically, the Jacobi constant is usually denoted $C = 2\,\Omega$.)

Surface Plot — (x,y)

Here is a procedure to evaluate the Jacobi constant:

$S := \mathrm{fn}(\mathrm{rhs}(JC), \mu, \xi, \eta, \zeta)$

$$S := (\mu, \xi, \eta, \zeta) \to \frac{1}{2}\xi^2 + \frac{1}{2}\eta^2 + \frac{1}{2}\zeta^2 + \frac{1-\mu}{\sqrt{\xi^2+2\,\xi\,\mu+\mu^2+\eta^2+\zeta^2}} + \frac{\mu}{\sqrt{\xi^2-2\,\xi+2\,\xi\,\mu+1-2\,\mu+\mu^2+\eta^2+\zeta^2}}$$

For plotting, the following two procedures are useful:

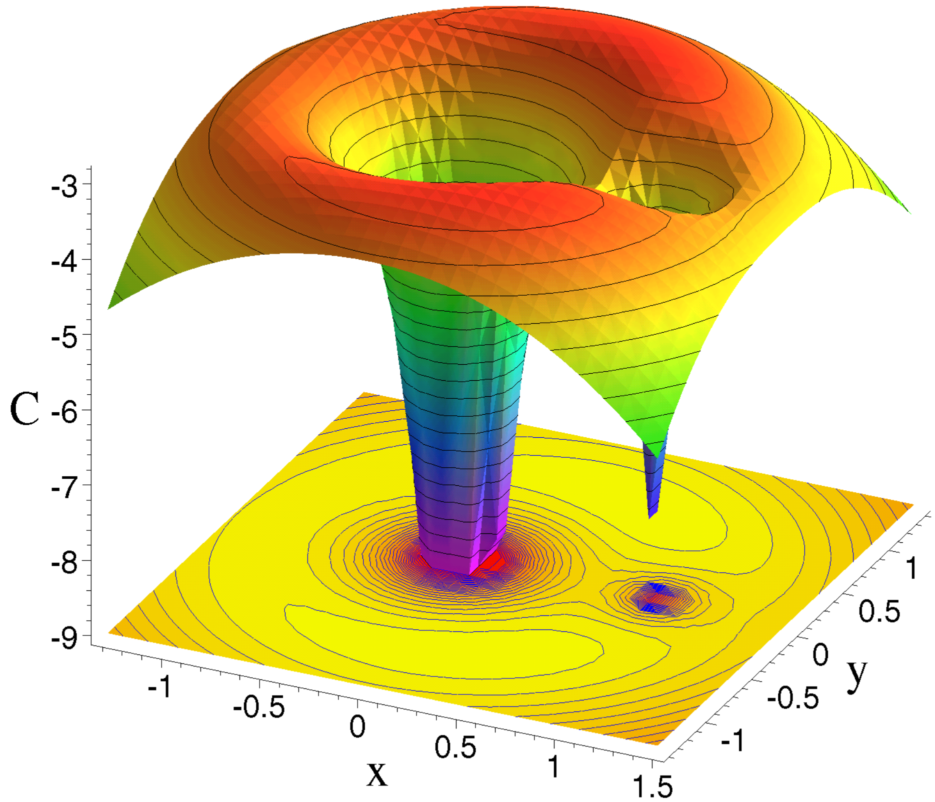
```
C := proc(Cmin, μ, ξ, η, ζ) local jc; jc := −S(μ, ξ, η, ζ); if jc < Cmin then jc := Cmin fi; jc end
```

```
Cplot := proc(μ, z, xmin, xmax, ymin, ymax, Cmin, n::posint, g::list, doscale)
local xy_scale, axistype, styletype, plt3D, plt2D, f, xform, disp, L, Cmax, cons, k;
```

```
    if doscale then xy_scale := Cmin / (xmax − xmin); axistype := NONE; styletype := PATCHNOGRID
    else xy_scale := 1; axistype := FRAMED; styletype := PATCHCONTOUR
    fi;
    L := light4;
    Cmax := −2.8;
    cons := [seq(Cmin + k*(Cmax − Cmin) / n, k = 0 .. n − 1)];
    plt3D := plot3d([xy_scale*x, xy_scale*y, 2*'C'(Cmin / 2, μ, x, y, z)], x = xmin .. xmax,
        y = ymin .. ymax, contours = cons, lightmodel = L, axes = axistype, grid = 'g', style = styletype);
    plt2D := plots[contourplot](2*'C'(Cmin / 2, μ, x, y, z), x = xmin .. xmax, y = ymin .. ymax, grid = 'g',
        contours = cons, filled = true);
    f := plottools[transform]((x, y) → [x, y, Cmin]);
    if doscale then plt3D
    else plots[display]({plt3D, f(plt2D)}, orientation = [−65, 60], projection = ORTHOGONAL,
        axes = axistype, lightmodel = L, labels = [x, y, C])
    fi
end
Warning, `Cmin` is a lexically scoped parameter
```

Here is a plot of the Jacobi surface for μ = 0.15 and z = 0:

```
Cplot(.15, 0, –1.3, 1.5, –1.4, 1.4, –9, 25, [30, 30], false)
```



Let's make a VRML object of the surface, suitable for viewing in a VRML browser such as Cosmo (see http://cosmo.sgi.com/):

$plottools_{vrml}(\mathrm{Cplot}(.15, 0, -1.3, 1.5, -1.4, 1.4, -9, 25, [35, 35], true),$
$\text{"d:/dynamics/ertb/CRTBSurfaceXY.wrl"}, background\_color = black, shininess = 1,$
$smooth\_shading = true)$

Surface Plot — (R,θ)

Define $R = \sqrt{\xi^2 + \eta^2 + \zeta^2}$ and convert to spherical polar coordinates.

$\Omega = \mathrm{map}(simplify, \mathrm{rhs}(JC), \{\xi^2 + \eta^2 + \zeta^2 = R^2\})$

$\mathrm{subs}(\xi = R\cos(\phi)\sin(\theta), \%)$

$$\Omega = \frac{1}{2}R^2 - \frac{-1+\mu}{\sqrt{2\,R\cos(\phi)\sin(\theta)\,\mu + \mu^2 + R^2}} + \frac{\mu}{\sqrt{\mu^2 - 2\,\mu + 1 + (-2+2\,\mu)\,R\cos(\phi)\sin(\theta) + R^2}}$$

JCpolar := %

Spolar := fn(rhs(*JCpolar*), μ, *R*, θ, φ)

$$Spolar := (\mu, R, \theta, \phi) \rightarrow \frac{1}{2}R^2 - \frac{-1+\mu}{\sqrt{2\,R\cos(\phi)\sin(\theta)\,\mu + \mu^2 + R^2}} + \frac{\mu}{\sqrt{\mu^2 - 2\,\mu + 1 + (-2+2\,\mu)\,R\cos(\phi)\sin(\theta) + R^2}}$$

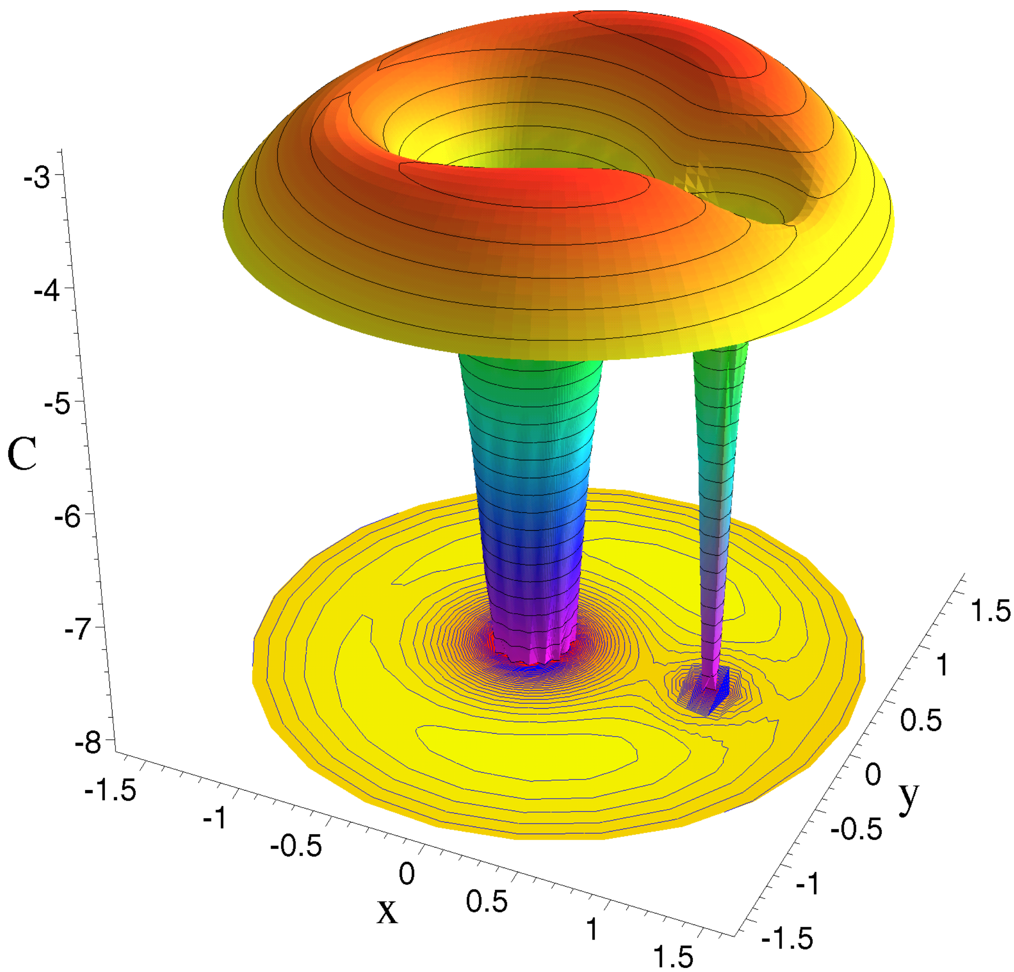
```
Cpolar := proc(Cmin, μ, R, θ, φ) local jc; jc := -Spolar(μ, R, θ, φ); if jc < Cmin then jc := Cmin fi; jc end

readlib(addcoords)(z_cylindrical, [z, r, φ], [r cos(φ), r sin(φ), z])

Cpolarplot := proc(F::procedure, μ, θ, Rmin, Rmax, Cmin, n::posint, g::list, doscale)
local Rscale, axistype, styletype, plt3D, plt2D, f, xform, disp, L, cons, Cmax, k;
    if doscale then
        Rscale := 1*Cmin / (2*(Rmax - Rmin)); axistype := NONE; styletype := PATCHNOGRID
    else Rscale := 1; axistype := FRAMED; styletype := PATCHCONTOUR
    fi;
    L := light4;
    Cmax := -2.8;
    cons := [seq(Cmin + k*(Cmax - Cmin) / n, k = 0 .. n - 1)];
    plt3D := plot3d([2*'F'(Cmin / 2, μ, R, θ, φ), R*Rscale, φ], R = Rmin .. Rmax, φ = 0 .. 2*π,
        coords = z_cylindrical, contours = cons, lightmodel = L, axes = axistype, grid = 'g',
        style = styletype);
    plt2D := plots[contourplot]([2*'F'(Cmin / 2, μ, R, θ, φ), R*Rscale, φ], R = Rmin .. Rmax,
        φ = 0 .. 2*π, coords = z_cylindrical, contours = cons, filled = true);
    f := plottools[transform]((x, y) → [x, y, Cmin]);
    if doscale then plt3D
    else plots[display]({plt3D, f(plt2D)}, orientation = [-65, 60], projection = ORTHOGONAL,
        axes = axistype, lightmodel = L, labels = [x, y, C])
    fi
end
Warning, `\`Cmin\` is a lexically scoped parameter`
```

Here is a plot of the Jacobi surface for μ = 0.15 and inclination angle 0 degrees:

$$\text{Cpolarplot}\left(Cpolar, .15, \frac{\pi}{2}, 0, 1.6, -8, 30, [30, 120], false\right)$$



Let's make another VRML object:

$$plottools_{vrml}\left(\text{Cpolarplot}\left(Cpolar, .15, \frac{\pi}{2}, 0, 1.6, -6.5, 30, [30, 60], true\right),\right.$$
$$\left."\text{d:/dynamics/ertb/CRTBSurface.wrl}", background\_color = black, shininess = 1, smooth\_shading = true\right)$$

Expansion on Distance from m_1

Translate the coordinate origin to the primary mass, m_1.

$\text{subs}(\xi = \xi - \mu, JC)$

$\Omega = \mathrm{map}(\mathit{simplify}, \mathrm{rhs}(\%), \{\xi^2 + \eta^2 + \zeta^2 = R^2\})$

$\Omega = \mathrm{map}(\mathit{simplify}, \mathrm{rhs}(\%), \mathit{assume} = \mathit{positive})$

$\mathrm{subs}(\xi = R \sin(\theta)\cos(\phi), \%)$

$$\Omega = -R \sin(\theta)\cos(\phi)\,\mu + \frac{1}{2}\mu^2 + \frac{1}{2}R^2 - \frac{-1+\mu}{R} + \frac{\mu}{\sqrt{-2\,R\sin(\theta)\cos(\phi) + 1 + R^2}}$$

Now expand on R and on $\varepsilon = \frac{\pi}{2} - \theta$, thereby approximating a nearly coplanar orbit around the primary mass orbited by a distant secondary mass.

$\mathrm{collect}\left(\mathrm{expansion}\left(\mathrm{subs}\left(\theta = \frac{\pi}{2} - \varepsilon, \%\right), R, 5, \varepsilon, 2\right), [R, \mu]\right)$

$\Omega = \mathrm{collect}(\mathrm{map}(\mathit{combine}, \mathrm{rhs}(\%), \mathit{trig}), [R, \mu, \cos, \varepsilon])$

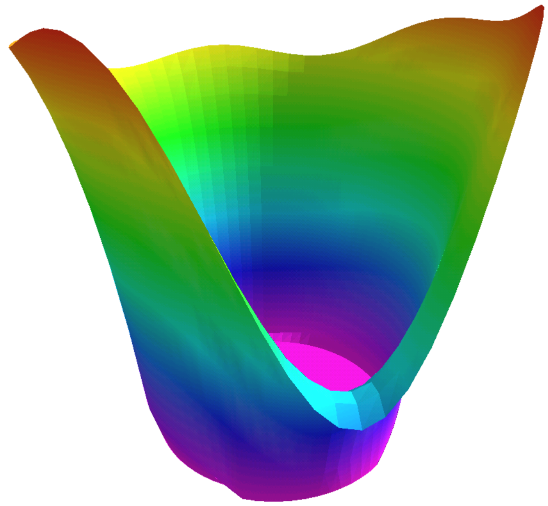
$\mathit{eqn} := \%$

$$\Omega = \left(\left(-\frac{315}{256}\varepsilon^2 + \frac{63}{128}\right)\cos(5\,\phi) + \left(-\frac{435}{128}\varepsilon^2 + \frac{15}{64}\right)\cos(\phi) + \left(-\frac{735}{256}\varepsilon^2 + \frac{35}{128}\right)\cos(3\,\phi)\right)\mu\,R^5$$
$$+ \left(\left(-\frac{5}{2}\varepsilon^2 + \frac{5}{16}\right)\cos(2\,\phi) + \left(-\frac{35}{32}\varepsilon^2 + \frac{35}{64}\right)\cos(4\,\phi) + \frac{9}{64} - \frac{45}{32}\varepsilon^2\right)\mu\,R^4$$
$$+ \left(\left(\frac{3}{8} - \frac{33}{16}\varepsilon^2\right)\cos(\phi) + \left(\frac{5}{8} - \frac{15}{16}\varepsilon^2\right)\cos(3\,\phi)\right)\mu\,R^3$$
$$+ \left(\left(\left(-\frac{3}{4}\varepsilon^2 + \frac{3}{4}\right)\cos(2\,\phi) - \frac{3}{4}\varepsilon^2 + \frac{1}{4}\right)\mu + \frac{1}{2}\right)R^2 + \mu + \frac{1}{2}\mu^2 + \frac{1-\mu}{R}$$

$\mathit{Sx} := \mathrm{fn}(\mathrm{rhs}(\mathit{eqn}), \mu, R, \varepsilon, \phi)$

$\mathit{Cx} := \mathbf{proc}(\mathit{Cmin}, \mu, R, \varepsilon, \phi)\ \mathbf{local}\ \mathit{jc};\ \mathit{jc} := -\mathrm{Sx}(\mu, R, \varepsilon, \phi);\ \mathbf{if}\ \mathit{jc} < \mathit{Cmin}\ \mathbf{then}\ \mathit{jc} := \mathit{Cmin}\ \mathbf{fi};\ \mathit{jc}\ \mathbf{end}$

$\mathrm{Cpolarplot}(\mathit{Cx}, .15, 0, 0, .8, -4, 25, [20, 72], \mathit{true})$



save/restart

save "d:/dynamics/ertb/CRTBSurfaces.m"

restart

read "d:/dynamics/ertb/CRTBSurfaces.m"

readlib(*addcoords*)(*z_cylindrical*, [*z*, *r*, ϕ], [*r* cos(ϕ), *r* sin(ϕ), *z*])

?