Binary Orbits

Daniel George - dgeorge5@illinois.edu

Finding minimum mass of planet

Defining parameters and constants

```
G = Quantity["GravitationalConstant"];
mS = Quantity[1., "SolarMass"];(*Mass of star*)
rS = Quantity[1., "SolarRadius"]; (*Radius of star*)
oP  = Quantity[40., "Days"]; (*Period of orbit*)
vR  = Quantity[10., "m/s"](*Observed radial velocity*)
```

Solving functions of inclination angle (a)

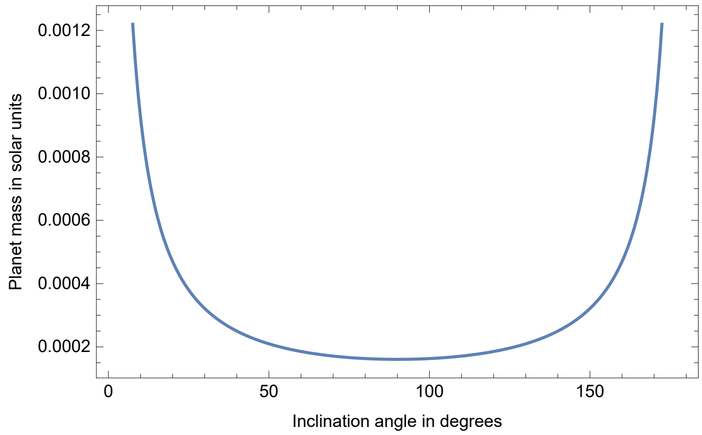
We solve the distances to the two bodies from the center of mass of the system, as well as the mass and velocity of the second body. This solution is accurate for any two bodies, regardless of their mass ratio.

```
In[60]:= {dS[i_], dP[i_], mP[i_], vP[i_]} =
  Assuming[0 < i < Pi, Block[{dS, dP, mP, vP,
      vS = vR / Sin[i] },
     Quiet@Solve[ (*Equations*)

        {2 Pi dS / oP == vS(*Time period of star*),
         2 Pi dP / oP == vP(*Time period of planet*),
         vS mS == mP vP(*Conservation of momentum*),
         mP * vP^2 / dP == G mS mP / (dP + dS)^2(*Centripetal force equals gravity*),

         0 < i < Pi(*Range of angle*)},
        {dS, dP, mP, vP}, Reals] // Simplify][[1, All, 2]]];
```

Plot of planet mass vs angle

In[63]:=
```
Plot[mP[i °], {i, 0, 180}, Frame → True,
 FrameLabel → {"Inclination angle in degrees", "Planet mass in solar units"}]
```

Out[63]=



Therefore the minimum mass is at the edge-on case (i = 90°).

In[28]:=
```
UnitConvert[mP[Pi / 2], "JupiterMass"]
```

Out[28]= 0.16829154877683 $M_{♃}$

Values for the edge-on case

In[30]:=
```
{dS[Pi / 2], dP[Pi / 2], mP[Pi / 2], vP[Pi / 2]} // Column
```

Out[30]=

$5.5003948332559 \times 10^{6}$ m

$3.4238679276079 \times 10^{10}$ m

0.00016064856909065 $M_{\odot}$

62 247.675510618 m / s

Solving angle range for crossing

Given density of planet

In[53]:=
```
rhoP = Jupiter (planet) ["Density"];
```

Assuming that the tangent touches both bodies

In[55]:=
```
{iMax, x, y, rP} = Module[{i, x, y, rP},
   List @@ Quiet@Reduce[(*Equations*)

      {Abs@Cos[i] == rS / (y) == rP / (x),
        (x + y) == dP[i] + dS[i],
        mP[i] / (4 / 3 Pi (rP) ^3) == rhoP,

        0 < i < Pi / 2, x > rP, y > rS}
      /. q_Quantity :> QuantityMagnitude[q, "SI"], {i, x, y, rP}, Reals][[All, 2]]];
```

The range of orbital inclinations in degrees

In[48]:=
```
{iMax, Pi - iMax} / °
```

Out[48]= {89.93425608962, 90.06574391038}

Radius of planet

In[58]:=
```
Quantity[rP, "Meters"]
```

Out[58]= $3.8597896555509 \times 10^{7}$ m