Gravitation

PreLab submission with a pass grade is required to begin the lab.
Must be submitted no later than right before the start of the lab.

Name: Aryan Malhotra
Section: H4
Date: 11/13/2024

Readings

You can review the concepts using Wikipedia or your favorite textbook,
Gravitation, Escape velocity, Kepler's Law.

Gravitation, Force & Energy

Newton's law of Universal Gravitation tells us that the gravitational attraction between two masses, m and M, is of magnitude

(1) $F = \frac{G\,m\,M}{r^2}$

where G = $6.67 \times 10^{-11} \mathrm{N} m^2\ \mathrm{kg}^{-2}$ and r is their radial separation. The direction of the force is towards the other mass (radial and attractive).

The potential energy, U, of mass m due to the gravitational attraction of M is given by,

(2) $U = -G \frac{m\,M}{r}$

If the orbit is **circular** the relationship between the speed of m about M and the distance R between m and M can be easily derived from Newton's second law,

(3) $F = \frac{G\,m\,M}{r^2} = m\,a = \frac{m\,v^2}{r}$,

(4) $v^2\,r = G\,M$.

Using this gravitational force one can solve the equation of motion and find that the orbits can be any one of the conic sections, i.e. **circle**, **ellipse**, **parabola**, or **hyperbola**. So the two body problem, subject to a $1/r^2$ force between them, is exactly solvable and has analytical solutions.

Dialog:

In[5]:=
```
MEarth = PlanetData["Earth", "Mass"]
```

Out[5]= 5.97×10^{24} kg

In[4]:=
```
MSun = StarData["Sun", "Mass"]
```
Out[4]= 1.988×10^{30} kg

In[3]:=
```
G = 6.67 * 10 ^ (-11)
```
Out[3]= 6.67×10^{-11}

Escape velocity from the Sun Vscp(Sun)

- A spaceship of mass m=150 kg is passing through solar system. When it is 1.00×10^6 km away from the center of the Sun, what should it's velocity be to escape the Sun's gravity?

U = GMm/r

For escape, this has to be equal to 1/2 mv^2

It must be at least 515.027 m/s (Here, total kinetic energy overcomes the potential energy)

In[10]:=
```
VescSun = N[Sqrt[2 * G * MSun[[1]] / (1.00 * 10^9)]]
```
Out[10]=

515 027.

Circular orbit around the Sun Vorb(Sun)

- When it is 1.00×10^6 km away from the Sun, what velocity does it need to have to be in a circular orbit around the Sun?

For an orbit, the gravitational force GMm/r^2 has to be the centripetal force mv^2/r.

Hence it must have 364179m/s of velocity to be in a stable orbit at the same distance away. (Here, potential energy is twice the negative of the kinetic energy)

In[11]:=
```
VorbitSun = N[Sqrt[G * MSun[[1]] / (1.00 * 10^9)]]
```
Out[11]=

364 179.

Escape velocity from the Earth Vscp(Earth)

- Calculate the spaceship's escape velocity from Earth's gravity when it is 1.00×10^4 km away from the center of the Earth.

The escape velocity would be 282257m/s for the given conditions.

In[12]:=
```
VescEarth = N[Sqrt[2 * G * MEarth[[1]] / (1.00 * 10^4)]]
```
Out[12]=

282 257.

Circular orbit around the Earth Vorb(Earth)

- What velocity does it need to have to be in a circular orbit around the Earth, when it is 1.00×10^4 km away from the planet?

The orbital velocity at the given constraints would be about 199586 m/s.

In[13]:=
```
VorbitEarth = N[Sqrt[G * MEarth[[1]] / (1.00 * 10^4)]]
```
Out[13]=

199 586.

Results

- Record your results in the table below.

In[14]:=
```
Grid[{{Text["Gravitaion"], SpanFromLeft},
  {"Vscp(Sun), m/s", "Vorb(Sun), m/s", "Vscp(Earth), m/s", "Vorb(Earth), m/s"},
  {VescSun, VorbitSun, VescEarth, VorbitEarth}}, Frame → All]
```
Out[14]=

Gravitaion			
Vscp(Sun), m/s	Vorb(Sun), m/s	Vscp(Earth), m/s	Vorb(Earth), m/s
515 027.	364 179.	282 257.	199 586.

Verifying Results Using Numerical Analysis

To get familiar with the lab simulations and verify your answers above, use the Simulation Cell 0 below. You will only need or might need to change the lines of the code which are highlighted in light red color.

Warnings:

- After opening this nb file run the Simulation Cell (denoted by orange bar) to avoid running through graphic boxes full of errors.

- NDSolveValue might give you an accuracy warning. You can safely ignore it.

- If the "Dynamic Updating" crashes and does not let you to use the Manipulation, reset it. See Evaluation > Dynamic Upgrading Enabled option from the top menu. Uncheck-mark and check-mark this option, or click on Abort on the pop-up message and go and check-mark the option.

- Write in one short paragraph (less than 30 words) which parameters you changed and what you have observed.

Answer:

I first changed x[0] to 10^8 (I assume this makes the object start off at a very far away position). Since it still had the same velocity, the object escaped the gravity. The potential energy Asymptotically decayed to 0 over time.

I changed the y[0] from 10^7 to 100. It just broke the software (I'm guessing because The energies would blow up to infinity, as scene in a graph that showed up for a second)
Changing the mass to an order of 10^30 kgs seemed to have crunched the entire orbital path into a straight line with again, an every increasing potential energy.
changing it to 10^23.9 was a sweet spot where the orbit was a very elongated ellipse.
Changing the Max time just increased simulation duration.

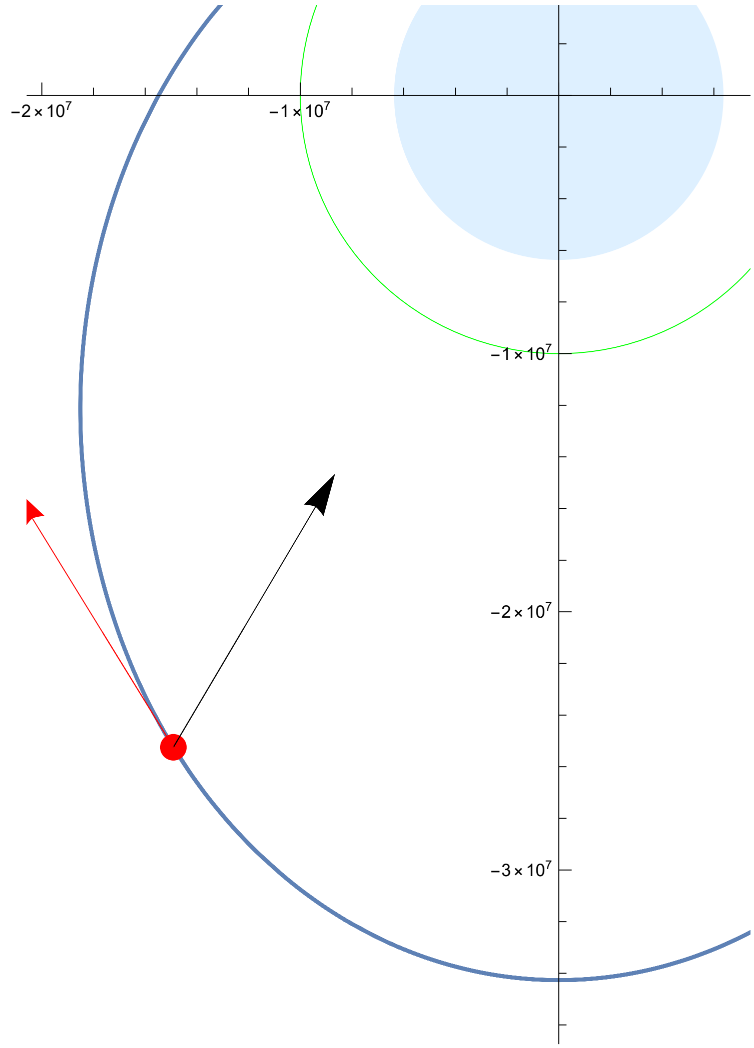
Simulation Cell 0

In[243]:=

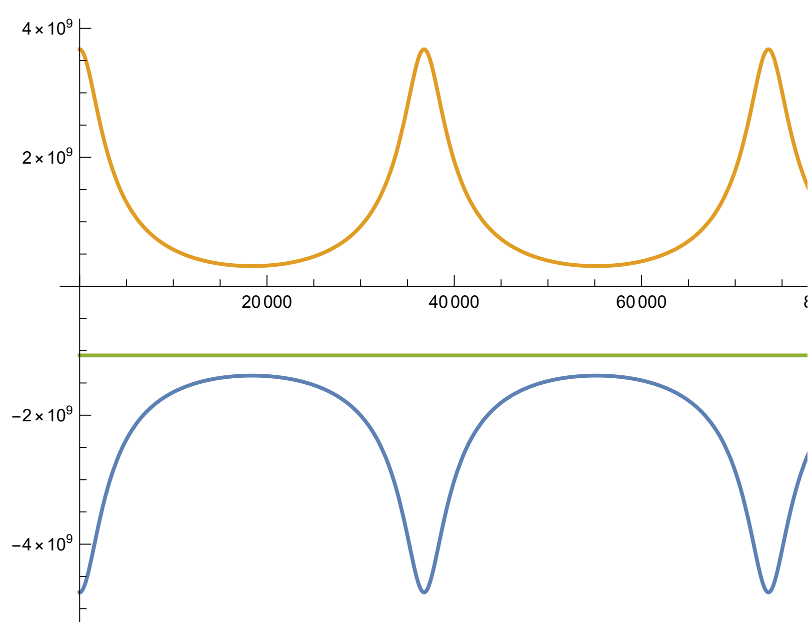
```
(* --- The chunk of code C1 begins. --- *)
PE[x_, y_] = -G M m / (x^2 + y^2) ^ (1 / 2);
KE[vx_, vy_] = (1 / 2) m (vx^2 + vy^2);
Energy[x_, y_, vx_, vy_] = PE[x, y] + KE[vx, vy];
(* End of chunk C1. *)

(* --- The chunk of code C2 begins. --- *)
G = 6.672 * 10^-11; (* Gravitational constant in m³kg⁻¹s⁻². *)
M = 5.972 * 10^23.9; (* Mass of Earth in kg. You can change this for the Sun. *)
m = 150; (* Mass of the object in kg. *)
R = 6.371 * 10^6; (* Earth mean radius in m. *)
(* End of chunk C2. *)

(* --- The chunk of code C3 begins. --- *)
tmax2 = 100 000; (* Max1mum running time *)
(* Below we are solving the equations of motion for an object much smaller than Earth
 (or Sun if you change M above) in its gravitational field with given initial conditions. *)
(* x2 and y2 are the solutions for the position of the object. ii stands for part ii. *)
{x2, y2} = NDSolveValue[{
    x''[t] == -((G M x[t]) / (x[t]^2 + y[t]^2) ^ (3 / 2)),
    y''[t] == -((G M y[t]) / (x[t]^2 + y[t]^2) ^ (3 / 2)),
    (* Set the initial conditions based
     on your calculations to get a circular motion or escape. *)
    x[0] == 0, (* Let us keep this 0 m. *)
    y[0] == 10^7, (* Initial distance from Earth/Sun in m *)
    x'[0] == 7000, (* Initial velocity in x direction in m/s. *)
    y'[0] == 0, (* Initial velocity in y direction in m/s. *)
    WhenEvent[x[t]² + y[t]² - R² < 0, {tmax2 = t, "StopIntegration"}]},
   {x, y}, {t, 0, tmax2}, MaxSteps → 10 000 000];
(* End of chunk C3. *)

scaleAccPos = Abs[x2[tmax2 / 2] / (Sqrt[x2''[tmax2 / 2]^2 + y2''[tmax2 / 2]^2])];
(* How big the acceleration vector shown on position graph. *)
scaleVelPos = Abs[x2[tmax2 / 2] / (Sqrt[x2'[tmax2 / 2]^2 + y2'[tmax2 / 2]^2])];
(* How big the velocity vector shown on position graph. *)

(* Below we are using Manipulate command with an interactive
 slider t to change the time and have an interactive simulation. *)
```

```
Manipulate[
 Grid[
  {
   {Show[
     ParametricPlot[{x2[t], y2[t]}, {t, 0, tmax2},
      AxesLabel → {"x", "y"}, PlotStyle → Automatic, PlotRange → Full, ImageSize → Large],
     Graphics[{LightBlue, Disk[{0, 0}, R]}, ImageSize → Large, PlotRange → All],
     Graphics[{Green, Circle[{0, 0}, 10^7]}], (* A Green Circle as a guide,
     at r=10^7m for Earth's case and r=10^9m for the Sun's case. *)
     Graphics[{PointSize[.025], Hue[0], Point[{x2[t], y2[t]}]}],
     Graphics[{Black,
       Arrow[{{x2[t], y2[t]}, {x2[t] + scaleAccPos x2''[t], y2[t] + scaleAccPos y2''[t]}}]}],
     Graphics[
      {Red, Arrow[{{x2[t], y2[t]}, {x2[t] + scaleVelPos x2'[t], y2[t] + scaleVelPos y2'[t]}}]}]
    ],
    Grid[{
      {Text["TABLE X01. Measurements"], SpanFromLeft},
      {"Time, s", "r, m", "v, m/s", "a, m/s^2", "K.E., J", "P.E., J", "E, J"},
      {t, Sqrt[x2[t]^2 + y2[t]^2], Sqrt[x2'[t]^2 + y2'[t]^2], Sqrt[x2''[t]^2 + y2''[t]^2],
       KE[x2'[t], y2'[t]], PE[x2[t], y2[t]], Energy[x2[t], y2[t], x2'[t], y2'[t]]},
      {Text["TABLE X02. xy Notation Variables"], SpanFromLeft},
      {"Time, s", "x, m", "y, m", "vx, m/s", "vy, m/s", "ax, m/s^2", "ay, m/s^2"},
      {t, x2[t], y2[t], x2'[t], y2'[t], x2''[t], y2''[t]}
     }, Frame → All]
   },
   {Show[
     Plot[{PE[x2[t], y2[t]], KE[x2'[t], y2'[t]], Energy[x2[t], y2[t], x2'[t], y2'[t]]},
      {t, 0, tmax2}, ImageSize → Large, PlotRange → All,
      PlotLegends → {"PE", "KE", "E"}, AxesLabel → {"Time (s)", ""}],
     Graphics[{PointSize[.025], Hue[0], Point[{t, PE[x2[t], y2[t]]}]}],
     Graphics[{PointSize[.025], Hue[0], Point[{t, KE[x2'[t], y2'[t]]}]}],
     Graphics[{PointSize[.025], Hue[0], Point[{t, Energy[x2[t], y2[t], x2'[t], y2'[t]]}]}]],
    Show[
     Graphics[{PointSize[0.015], Opacity[0.5], Point[{0, 0}]},
      AxesLabel → {Subscript["v", "x"], Subscript["v", "y"]}, Axes → True, ImageSize → Large],
     ParametricPlot[{x2'[t], y2'[t]}, {t, 0, tmax2}, PlotStyle → Automatic, PlotRange → Full],
     Graphics[{PointSize[.025], Hue[0], Point[{x2'[t], y2'[t]}]}, ImageSize → Large],
     Graphics[{Red, Arrow[{{0, 0}, {x2'[t], y2'[t]}}]}]
    ]
   }
  }], {t, 0, tmax2, Appearance → "Labeled"}]
```

Out[254]=



−2 × 10^7
−1 × 10^7
−1 × 10^7
−2 × 10^7
−3 × 10^7



4 × 10^9
2 × 10^9
20 000
40 000
60 000
−2 × 10^9
−4 × 10^9

Rutgers 275 Classical Physics Lab
“Gravitation”