Extension: Comparisons of the Apse Lines

- Averyl Xu
- Damla Temur
- Ian Gu
- Isaiah Liu
- Sambhu Ganesan

“All models are wrong, but some are useful.” - George Box

We extend on our previous work by evaluating how well three-body and four-body simulations predict the Moon’s orbit, focusing on apse line dynamics. The analysis compares apogee and perigee distances, apse lengths, and apsidal precession angles against real data. The results suggest that while making the model more complex improves the system in some ways, it has diminishing returns. Despite minor discrepancies, the three-body model already captures the general patterns of apse lengths and rotation well, with potential self-correcting errors.

First Attempts

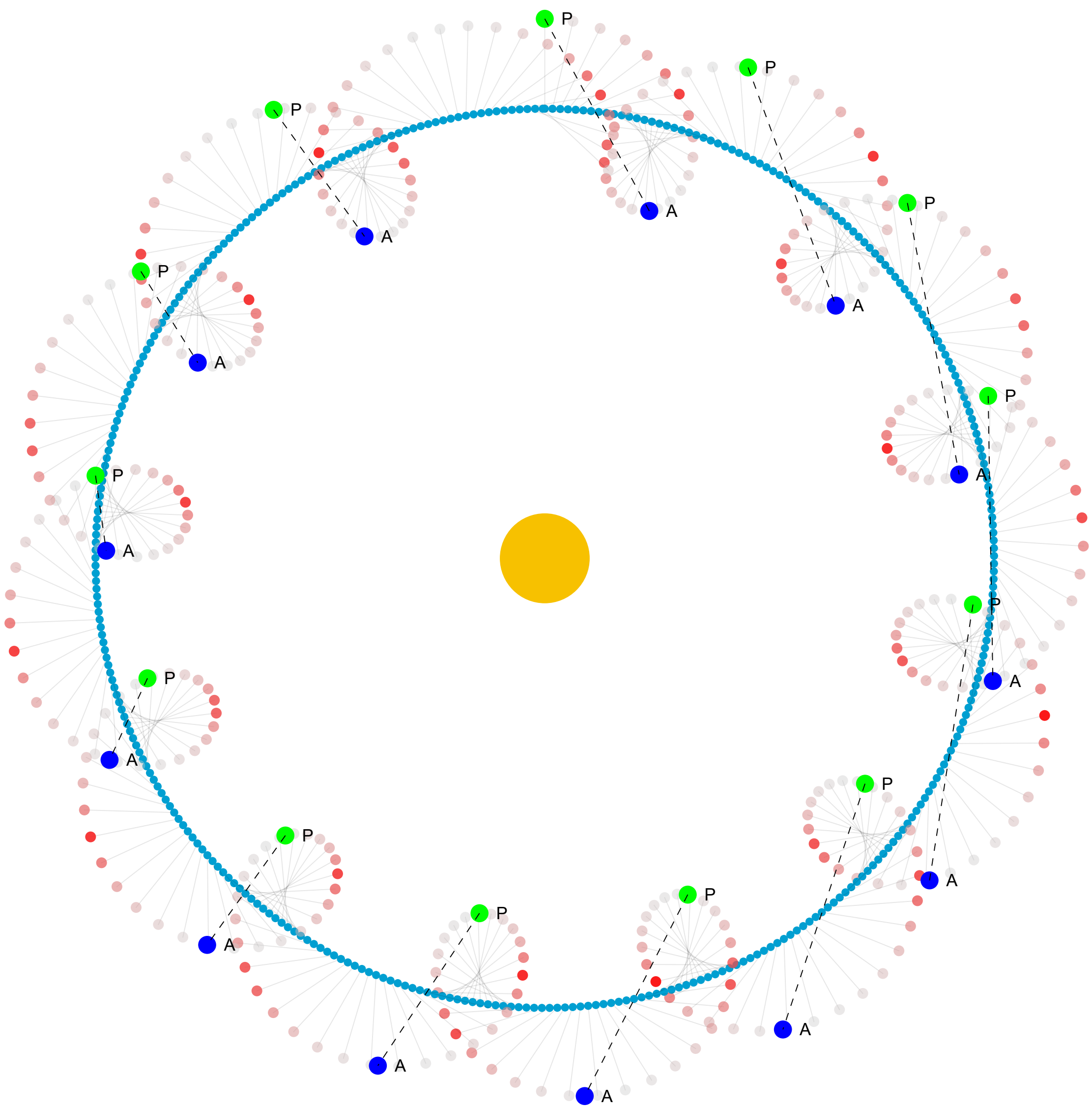
The moon's orbit is not perfectly circular; rather, it is shaped like an ellipse. As a result, due to the property of ellipses, there are times when the moon is closest to earth (aka supermoons) and times when the moon is the farthest and looks smallest (aka micromoons). Because of its elliptical orbit, we can also analyze the apse line: the line that connects the farthest point (apogee) with the closest point (perigee) in a lunar cycle.

In[351]:=

```
With[{dt = 1, apsisPeriod = 27.3, daysInYear = 365.25},
 Graphics[{Style[Disk[{0, 0}, .1], Hue[0.13, 1, 0.97]],
   Table[Module[{earthPos = {Sin[2 Pi t / daysInYear], Cos[2 Pi t / daysInYear]},
      moonPos = 0.2 {Sin[2 Pi t / apsisPeriod], Cos[2 Pi t / apsisPeriod]}},
     {Style[Line[{earthPos, earthPos + moonPos}], GrayLevel[0, .1]],
      Style[Disk[earthPos + moonPos, .012],
       Blend[{Red, GrayLevel[.8, .4]}, Sqrt[Abs[Cos[2 Pi (t + dt) / apsisPeriod]]]]],
      Style[Disk[earthPos, .009], Hue[0.54, 1, 0.82]]}], {t, 0, 365}],
   Table[Module[{tPerigee = n * apsisPeriod, tApogee = n * apsisPeriod + apsisPeriod / 2,
      earthPosPerigee, moonPosPerigee, earthPosApogee, moonPosApogee},
     earthPosPerigee = {Sin[2 Pi tPerigee / daysInYear], Cos[2 Pi tPerigee / daysInYear]};
     moonPosPerigee =
      0.2 {Sin[2 Pi tPerigee / apsisPeriod], Cos[2 Pi tPerigee / apsisPeriod]};
     earthPosApogee = {Sin[2 Pi tApogee / daysInYear], Cos[2 Pi tApogee / daysInYear]};
     moonPosApogee = 0.2 {Sin[2 Pi tApogee / apsisPeriod], Cos[2 Pi tApogee / apsisPeriod]};
     {Style[Disk[earthPosPerigee + moonPosPerigee, .02], Green],
      Text["P", earthPosPerigee + moonPosPerigee + {.05, 0}],
      Style[Disk[earthPosApogee + moonPosApogee, .02], Blue],
      Text["A", earthPosApogee + moonPosApogee + {.05, 0}],
      Style[Line[{earthPosPerigee + moonPosPerigee, earthPosApogee + moonPosApogee}],
       Black, Dashed]}], {n, 0, 12}],
   Module[{tPerigeeStart = 0, tPerigeeEnd = apsisPeriod * Floor[daysInYear / apsisPeriod],
     earthPosPerigeeStart, moonPosPerigeeStart, posStart, earthPosPerigeeEnd,
     moonPosPerigeeEnd, posEnd, angleShift}, earthPosPerigeeStart =
     {Sin[2 Pi tPerigeeStart / daysInYear], Cos[2 Pi tPerigeeStart / daysInYear]};
    moonPosPerigeeStart =
     0.2 {Sin[2 Pi tPerigeeStart / apsisPeriod], Cos[2 Pi tPerigeeStart / apsisPeriod]};
    posStart = earthPosPerigeeStart + moonPosPerigeeStart;
    earthPosPerigeeEnd =
     {Sin[2 Pi tPerigeeEnd / daysInYear], Cos[2 Pi tPerigeeEnd / daysInYear]};
    moonPosPerigeeEnd =
     0.2 {Sin[2 Pi tPerigeeEnd / apsisPeriod], Cos[2 Pi tPerigeeEnd / apsisPeriod]};
    posEnd = earthPosPerigeeEnd + moonPosPerigeeEnd;
    angleShift = ArcCos[(posStart.posEnd) / (Norm[posStart] Norm[posEnd])];
    Print["Calculated apsis shift: ", angleShift / Degree, " degrees"]]}]]
```

```
Calculated apsis shift: 8.50602 degrees
```

Out[351]=



In our first attempt to plot the apse lines, we assumed that the Earth was a perfectly circular orbit, and that the Moon exhibited a perfectly circular orbit too. Without the use of Newtonian physics, we plotted out first estimates of the motion of the apsis lines with parametric equations, polar geometry, and trigonometry. While this model is highly inaccurate, the patterns shown here reflect actual patterns with the simulations and the real data.

Setting Up Initial Conditions

We set up the initial conditions as we did in the last paper.

In[352]:=

```
sun = Sun STAR;
earth = Earth PLANET;
```

In[354]:=

```
sunMass = Sun STAR [ mass ];
earthMass = Earth PLANET [ mass ];
```

In[356]:=

```
dates = DateRange[Sat 1 Jan 2000 00:00:00 UTC,
   DatePlus[Sat 1 Jan 2000 00:00:00 UTC, UnitConvert[10 yr, "Seconds"]]];
```

In[357]:=

```
positionICRS[body_, date_] := UnitConvert[
   Quantity[AstroPosition[body, {"ICRS", "Date" → date}, "Cartesian"]["Data"],
    "AstronomicalUnit"], "Meters"];
initialSunPositionICRS1 = positionICRS[sun, Sat 1 Jan 2000 00:00:00 UTC];
initialEarthPositionICRS1 = positionICRS[earth, Sat 1 Jan 2000 00:00:00 UTC];
initialSunPositionICRS2 =
  positionICRS[sun, Sat 1 Jan 2000 00:00:00 UTC + 0.5 min];
initialEarthPositionICRS2 =
  positionICRS[earth, Sat 1 Jan 2000 00:00:00 UTC + 0.5 min];
initialSunVelocityICRS = (initialSunPositionICRS2 - initialSunPositionICRS1) /
   UnitConvert[0.5 min, "Seconds"];
initialEarthVelocityICRS = (initialEarthPositionICRS2 - initialEarthPositionICRS1) /
   UnitConvert[0.5 min, "Seconds"];
```

In[364]:=

```
normDifferences[position1_, position2_] :=
  Table[Norm[position1〚i〛 - position2〚i〛], {i, 1, Length[position2]}];
```

In[365]:=

```
sunPositionsICRS = Table[positionICRS[sun, i], {i, dates}];
earthPositionsICRS = Table[positionICRS[earth, i], {i, dates}];
```

In[367]:=

```
sunEarthDistancesICRS = earthPositionsICRS - sunPositionsICRS;
normDistancesEarthSunICRS = normDifferences[earthPositionsICRS, sunPositionsICRS];
```

In[369]:=

```
moon = Moon PLANETARY MOON ;
moonMass = Moon PLANETARY MOON [ mass ] ;
```

In[371]:=

```
initialMoonPositionICRS1 = positionICRS[moon, Sat 1 Jan 2000 00:00:00 UTC];
initialMoonPositionICRS2 =
  positionICRS[moon, Sat 1 Jan 2000 00:00:00 UTC + 0.5 min];
initialMoonVelocityICRS = (initialMoonPositionICRS2 - initialMoonPositionICRS1) /
   UnitConvert[0.5 min, "Seconds"];
moonPositionsICRS = Table[positionICRS[moon, i], {i, dates}];
```

In[375]:=

```
sunMoonDistancesICRS=moonPositionsICRS-sunPositionsICRS;
earthMoonDistancesICRS=earthPositionsICRS-moonPositionsICRS;
```

In[377]:=

```
normDistancesMoonSunICRS=normDifferences[moonPositionsICRS,sunPositionsICRS];
normDistancesMoonEarthICRS=normDifferences[moonPositionsICRS,earthPositionsICRS];
```

Apogee and Perigee

We defined the apogee to be the farthest point in a satellite's orbit to Earth and the Perigee to be the closest point in a satellite's orbit to Earth. We define the Apsis line as the line connected between the apogee and perigee of the Moon's orbit.

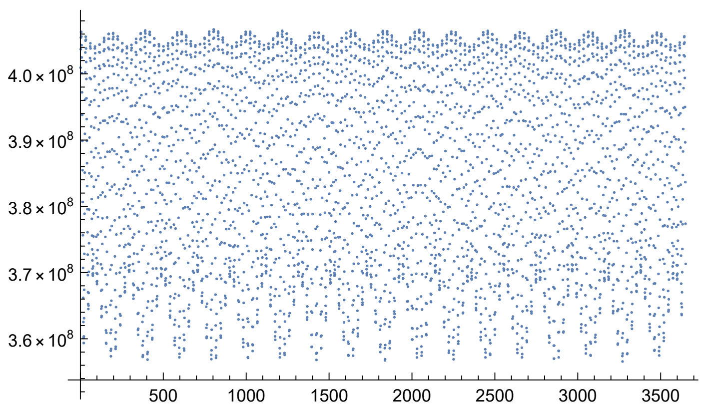
Apses Comparison

Another way to compare the simulations with real life data is to compare how their apses lines are different. We assume in this paper that the apsis line is defined by the line connecting the perigee to the next apogee. Below we have a plot of all the distances from the Moon to the Earth in the International Celestial Reference System (ICRS).

In[379]:=

```
ListPlot[normDistancesMoonEarthICRS]
```

Out[379]=

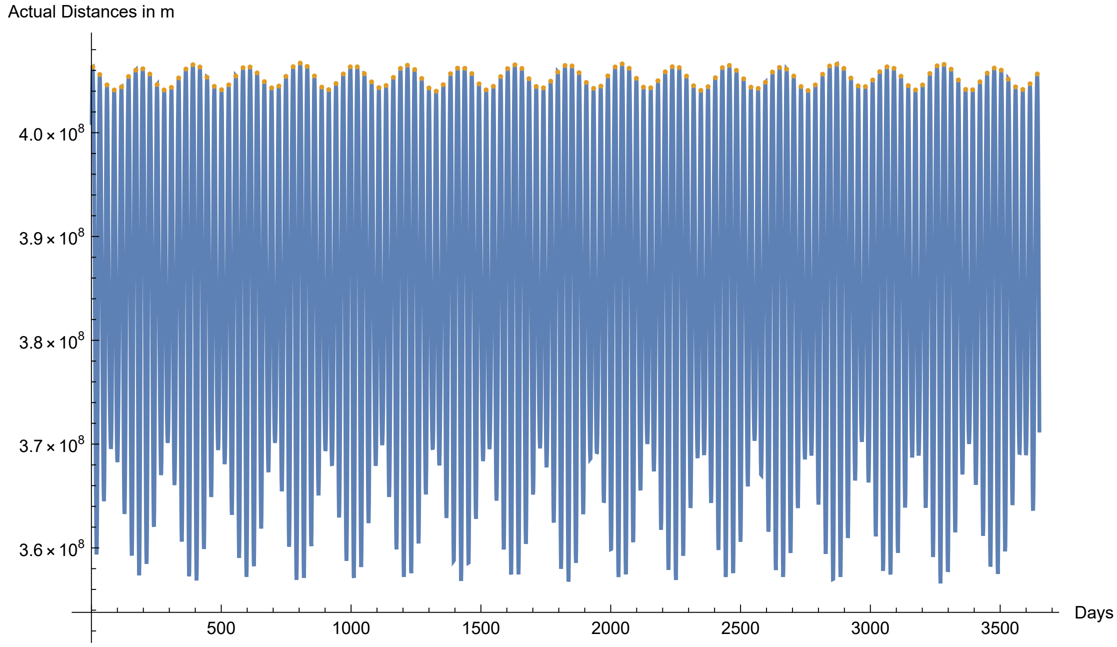


Now we will find the farthest point of the Moon, in it's orbit around Earth.

In[380]:=

```
peaks = FindPeaks[normDistancesMoonEarthICRS];
ListLinePlot[{normDistancesMoonEarthICRS, peaks},
 Joined → {True, False}, PlotStyle → {Automatic, PointSize[.005]},
 AxesLabel → {"Days", "Actual Distances in m"}, ImageSize → Large]
```

Out[381]=



Peaks (apogees) are stored as a pair: {day / index in the array, distance}

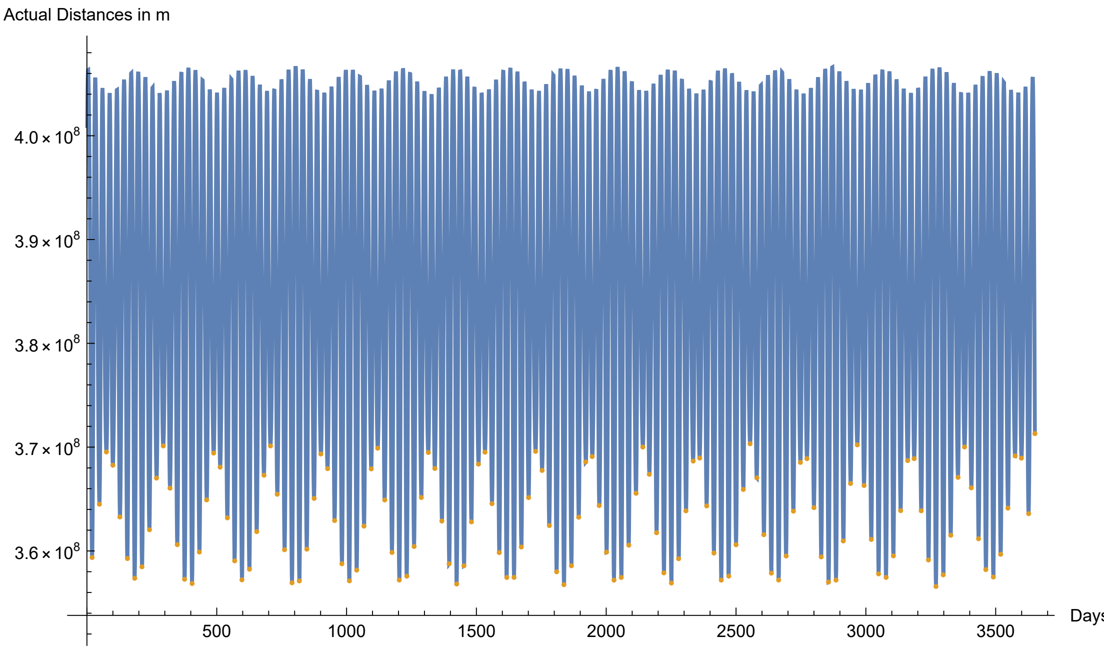
Yellow points in the plot below indicate the locations of the apogees. In this case, we use a simplified method to solve for the apogees: the local maxima of the distance of the Earth-Moon graph.

We use a similar method to find the locations of the matching perigees: the local minima in this case. This is equivalent to finding the local maxima of the negative function.

In[382]:=

```
valleys = Drop[FindPeaks[-normDistancesMoonEarthICRS] /. {x_, y_} :→ {x, -y}, 1];
ListLinePlot[{normDistancesMoonEarthICRS, valleys},
 Joined → {True, False}, PlotStyle → {Automatic, PointSize[.005]},
 AxesLabel → {"Days", "Actual Distances in m"}, ImageSize → Large]
```

Out[383]=



Valleys (perigees) are stored as a pair: {day / index in the array, distance}

In both plots, we see a sinusoidal pattern in the apogee and perigees, which matches theoretical prediction of small radial oscillations demonstrated by Nesterenko [arXiv:2310.05584]. The period is determined by the familiar, underlying harmonic oscillator second-order differential equation $x''(t) + \omega^2 x(t) = 0$ and has a solution $x(t) = C \cos(n t + \epsilon)$.

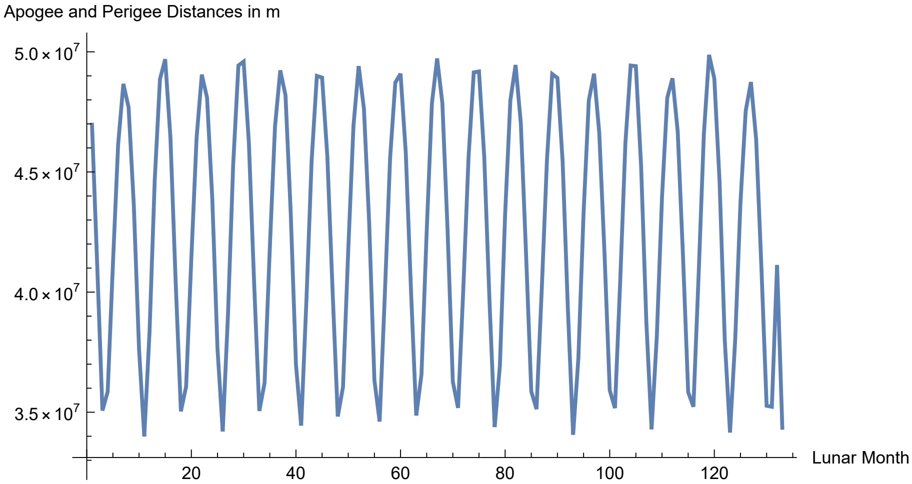
Interestingly though, the amplitude / variations of the apogee differences seem to be less than the variations of the perigees. Perhaps, this could be from the many complicated effects of the sun on the elliptical orbit (more information found in the NASA page).

The bottom graph shows the difference between the apse distance and perigee distance changes from line to line. There is a clear sinusoidal pattern also, which follows Newtonian patterns and predictions when the orbit of the Moon around the Earth is perturbed by the Sun.

In[384]:=

```
normEarthMoonApsesDifferences =
  Table[Abs[valleys[[i]][[2]] - peaks[[i]][[2]]], {i, Min[Length[valleys], Length[peaks]]}];
ListPlot[normEarthMoonApsesDifferences, Joined → True,
 AxesLabel → {"Lunar Month", "Differences in Apogee and Perigee Distances in m"}]
```

Out[385]=



Next, we compare the real data with the simulations from our last paper.

3 Body

We will build a 3-Body simulation of the Earth-Sun-Moon system.

In[386]:=

```
simulationData3BodyICRS = NBodySimulation["Newtonian", <|sun → <|"Mass" → sunMass,
      "Position" → initialSunPositionICRS1, "Velocity" → initialSunVelocityICRS|>,
    earth → <|"Mass" → earthMass,
      "Position" → initialEarthPositionICRS1, "Velocity" → initialEarthVelocityICRS|>,
    moon → <|"Mass" → moonMass, "Position" → initialMoonPositionICRS1 ,
      "Velocity" → initialMoonVelocityICRS|>|>, 10 yr, MaxSteps → Infinity];
simulationTime3BodyICRS = simulationData3BodyICRS["SimulationTime"]
```

Out[387]=

3.1536×10^8

Here we will get all the simulated data such as the positions and distances of the Sun, Earth, and the Moon.

In[388]:=

```
sunSimPos3BodyICRS =
  Table[Quantity[simulationData3BodyICRS[sun, "Position", t], "Meters"], {t, 0,
    simulationTime3BodyICRS, QuantityMagnitude[UnitConvert[1 day , "Seconds"] ✓]}];
earthSimPos3BodyICRS =
  Table[Quantity[simulationData3BodyICRS[earth, "Position", t], "Meters"],
   {t, 0, simulationTime3BodyICRS, QuantityMagnitude[UnitConvert[1 day , "Seconds"]]}];
moonSimPos3BodyICRS =
  Table[Quantity[simulationData3BodyICRS[moon, "Position", t], "Meters"], {t, 0,
    simulationTime3BodyICRS, QuantityMagnitude[UnitConvert[1 day , "Seconds"] ✓]}];
```

In[391]:=

```
sunEarthSimDistances3BodyICRS = earthSimPos3BodyICRS - sunSimPos3BodyICRS;
sunMoonSimDistances3BodyICRS = moonSimPos3BodyICRS - sunSimPos3BodyICRS;
earthMoonSimDistances3BodyICRS = earthSimPos3BodyICRS - moonSimPos3BodyICRS;
```

In[394]:=

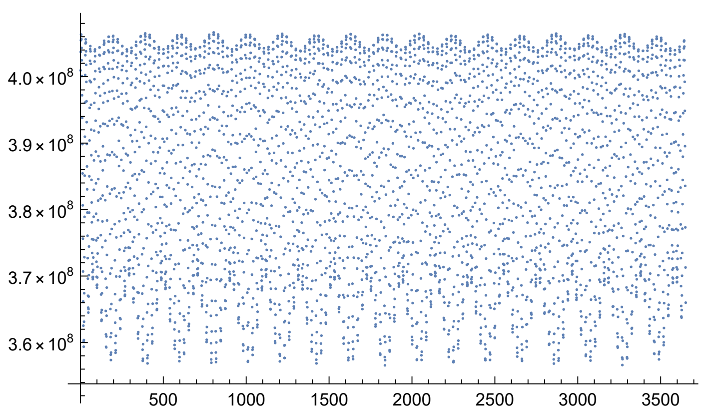
```
normSimDistancesEarthSun3BodyICRS =
  normDifferences[earthSimPos3BodyICRS, sunSimPos3BodyICRS];
normSimDistancesMoonSun3BodyICRS =
  normDifferences[moonSimPos3BodyICRS, sunSimPos3BodyICRS];
normSimDistancesMoonEarth3BodyICRS =
  normDifferences[earthSimPos3BodyICRS, moonSimPos3BodyICRS];
```

Similarly to our previous plots, we observe how the distances change from day to day.

In[397]:=

```
ListPlot[normSimDistancesMoonEarth3BodyICRS]
```

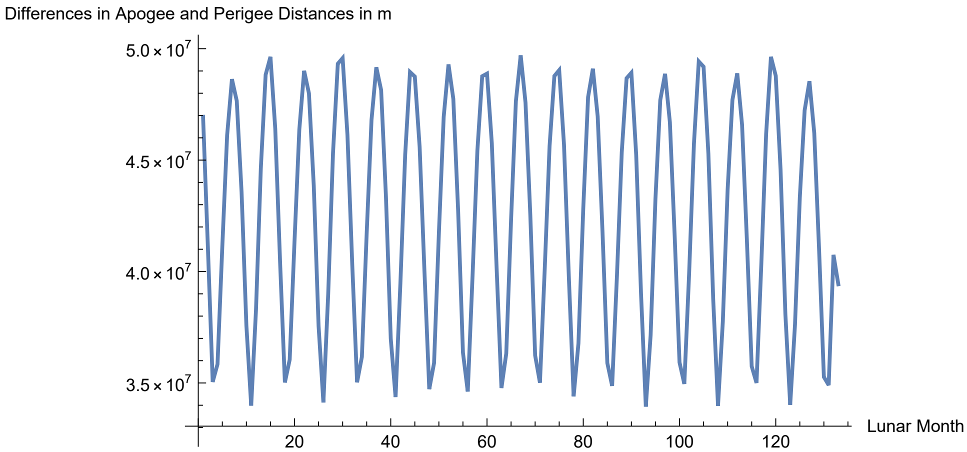
Out[397]=



In[398]:=

```
threeBodyPeaks = FindPeaks[normSimDistancesMoonEarth3BodyICRS];
threeBodyValleys =
  Drop[FindPeaks[-normSimDistancesMoonEarth3BodyICRS] /. {x_, y_} :→ {x, -y}, 1];
threeBodyNormEarthMoonApsesDistances =
  Table[Abs[threeBodyValleys[[i]][[2]] - threeBodyPeaks[[i]][[2]]],
   {i, Min[Length[threeBodyPeaks], Length[threeBodyValleys]]}];
ListPlot[threeBodyNormEarthMoonApsesDistances, Joined → True, AxesLabel →
  {"Lunar Month", "Differences in Apogee and Perigee Distances in m"}, ImageSize → 500]
```

Out[401]=

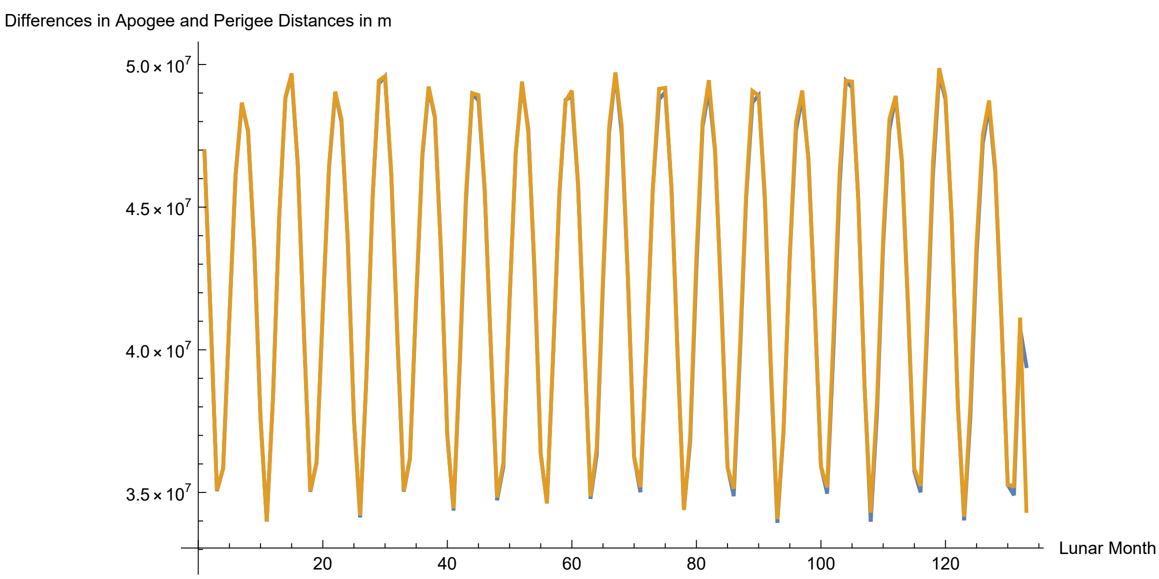


We can see a similar shape of the distance differences, also following a sinusoidal pattern with similar spikes and dips. We overlaid this graph with our original data below:

In[402]:=

```
ListPlot[{threeBodyNormEarthMoonApsesDistances, normEarthMoonApsesDifferences},
 Joined → True, ImageSize → 600,
 AxesLabel → {"Lunar Month", "Differences in Apogee and Perigee Distances in m"}]
```

Out[402]=



The points where the orange graph (our model) do not line up with the blue graph (actual data) as well occur much more often around the extrema. This may indicate that something else may be playing a role. We will consider Jupiter as a possible candidate.

4 Body

Now we will introduce Jupiter into the mix. We will build a 4-Body simulation of the Earth-Sun-Moon-Jupiter system.

In[403]:=

```
jupiter = Jupiter PLANET ;

jupiterMass = Jupiter PLANET [ mass ];

initialJupiterPositionICRS1 = positionICRS[jupiter, Sat 1 Jan 2000 00:00:00 UTC ];

initialJupiterPositionICRS2 =
  positionICRS[jupiter, Sat 1 Jan 2000 00:00:00 UTC + 0.5 min ];

initialJupiterVelocityICRS =
  (initialJupiterPositionICRS2 - initialJupiterPositionICRS1) /
   UnitConvert[0.5 min, "Seconds"];
```

In[408]:=

```
simulationData4BodyICRS = NBodySimulation["Newtonian", <|sun → <|"Mass" → sunMass,
      "Position" → initialSunPositionICRS1, "Velocity" → initialSunVelocityICRS|>,
    earth → <|"Mass" → earthMass, "Position" → initialEarthPositionICRS1,
      "Velocity" → initialEarthVelocityICRS|>, moon → <|"Mass" → moonMass,
      "Position" → initialMoonPositionICRS1 , "Velocity" → initialMoonVelocityICRS|>,
    jupiter → <|"Mass" → jupiterMass, "Position" → initialJupiterPositionICRS1,
      "Velocity" → initialJupiterVelocityICRS|>|>, 10 yr , MaxSteps → Infinity];

simulationTime4BodyICRS = simulationData4BodyICRS["SimulationTime"]
```

Out[409]=

3.1536×10^8

Just as before we will now collect the simulated data of the positions and distances.

In[410]:=

```
sunSimPos4BodyICRS =
  Table[Quantity[simulationData4BodyICRS[sun, "Position", t], "Meters"], {t, 0,
    simulationTime4BodyICRS, QuantityMagnitude[UnitConvert[1 day , "Seconds"]]}];
earthSimPos4BodyICRS =
  Table[Quantity[simulationData4BodyICRS[earth, "Position", t], "Meters"],
   {t, 0, simulationTime4BodyICRS, QuantityMagnitude[UnitConvert[1 day , "Seconds"]]}];
moonSimPos4BodyICRS =
  Table[Quantity[simulationData4BodyICRS[moon, "Position", t], "Meters"], {t, 0,
    simulationTime4BodyICRS, QuantityMagnitude[UnitConvert[1 day , "Seconds"]]}];
```

In[413]:=

```
sunEarthSimDistances4BodyICRS = earthSimPos4BodyICRS - sunSimPos4BodyICRS;
sunMoonSimDistances4BodyICRS = moonSimPos4BodyICRS - sunSimPos4BodyICRS;
earthMoonSimDistances4BodyICRS = earthSimPos4BodyICRS - moonSimPos4BodyICRS;
```

In[416]:=

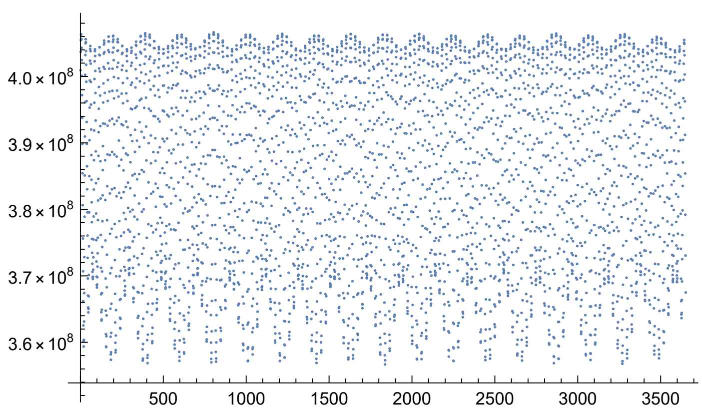
```
normSimDistancesEarthSun4BodyICRS =
  normDifferences[earthSimPos4BodyICRS, sunSimPos4BodyICRS];
normSimDistancesMoonSun4BodyICRS =
  normDifferences[moonSimPos4BodyICRS, sunSimPos4BodyICRS];
normSimDistancesMoonEarth4BodyICRS =
  normDifferences[earthSimPos4BodyICRS, moonSimPos4BodyICRS];
```

Similarly to our previous plots, we observe how the distances change from day to day.

In[419]:=

```
ListPlot[normSimDistancesMoonEarth4BodyICRS]
```

Out[419]=

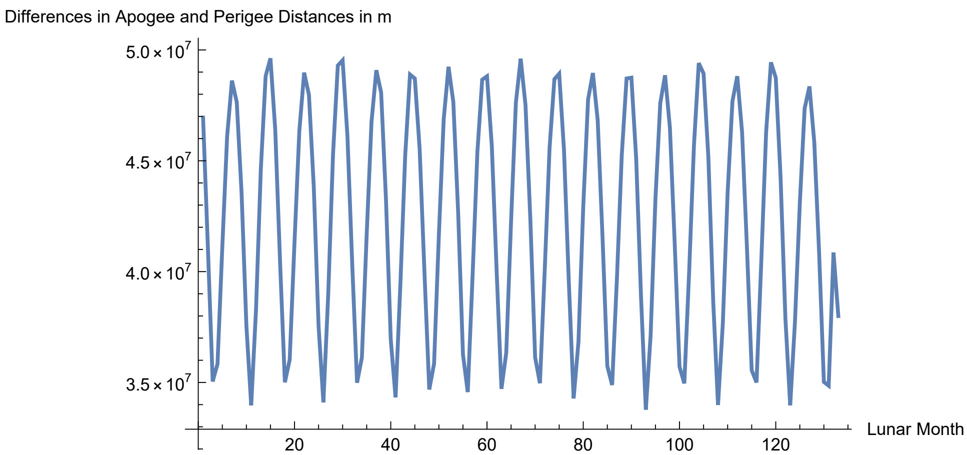


Once again we will find the peaks and valleys for the four body model as we did with the three body model. We will directly plot the differences in Apogee and Perigee distance below.

In[420]:=

```
fourBodyPeaks = FindPeaks[normSimDistancesMoonEarth4BodyICRS];
fourBodyValleys =
  Drop[FindPeaks[-normSimDistancesMoonEarth4BodyICRS] /. {x_, y_} :> {x, -y}, 1];
fourBodyNormEarthMoonApsesDistances =
  Table[Abs[fourBodyValleys[[i]][[2]] - fourBodyPeaks[[i]][[2]]],
   {i, Min[Length[fourBodyPeaks], Length[fourBodyValleys]]}];
ListPlot[fourBodyNormEarthMoonApsesDistances,
 AxesLabel → {"Lunar Month", "Differences in Apogee and Perigee Distances in m"},
 Joined → True, ImageSize → 500]
```

Out[423]=

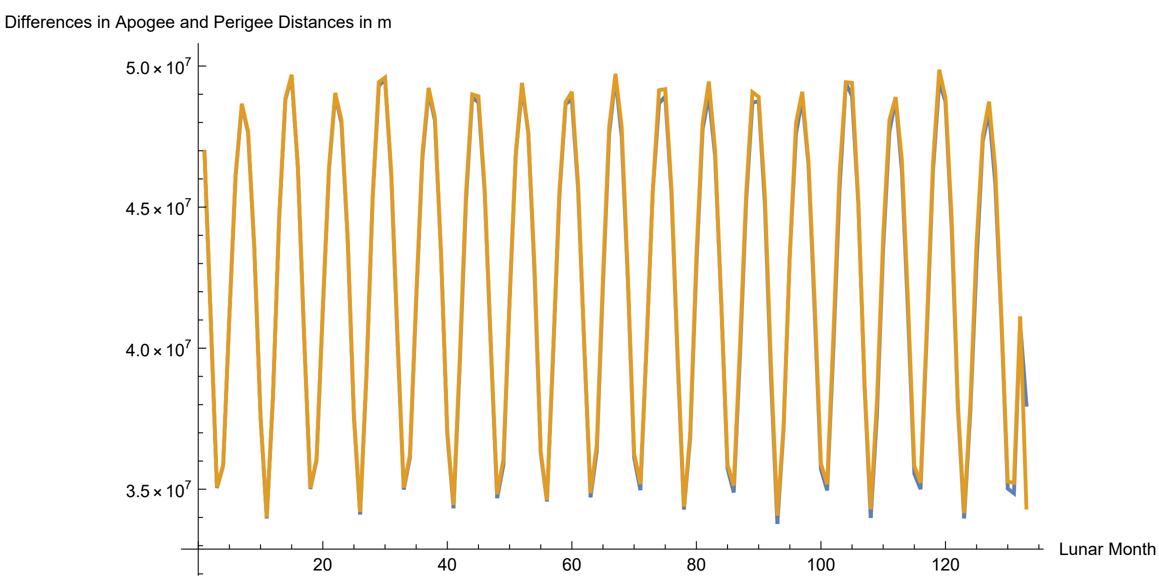


Below we will overlap our simulated data with the real data.

In[424]:=

```
ListPlot[{fourBodyNormEarthMoonApsesDistances, normEarthMoonApsesDifferences},
 Joined → True, AxesLabel →
  {"Lunar Month", "Differences in Apogee and Perigee Distances in m"}, ImageSize → 600]
```

Out[424]=



Visually, we can see that the four-body looks a bit more precise and consistent than the three-body. However, both plots are still very similar and yield relatively similar results. To better analyze this data, we found the variance of each simulation plot distance with the real distance.

In[425]:=

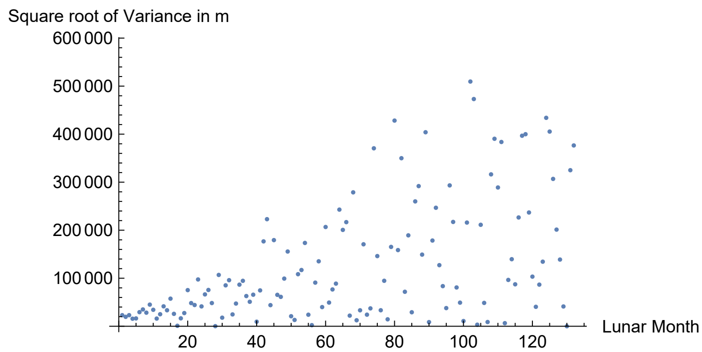
```
threeBodyApseVariations =
  Sqrt /@ ((threeBodyNormEarthMoonApsesDistances - normEarthMoonApsesDifferences) ^2);
fourBodyApseVariations =
  Sqrt /@ ((fourBodyNormEarthMoonApsesDistances - normEarthMoonApsesDifferences) ^2);
```

Now we can plot the variations below.

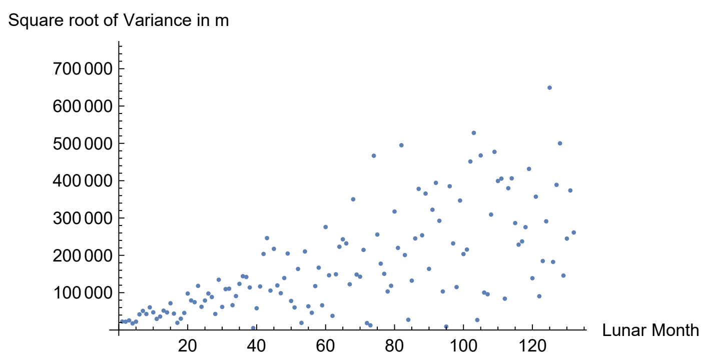
In[427]:=

```
ListPlot[threeBodyApseVariations,
 AxesLabel → {"Lunar Month", "Square root of Variance in m"}]
ListPlot[fourBodyApseVariations,
 AxesLabel → {"Lunar Month", "Square root of Variance in m"}]
```

Out[427]=



Out[428]=



It is also nice to have a numerical comparisons to see how different the two simulations ares. So lets compare the amount of errors of the simulations. To do this we will first collect the number of data points in each model.

In[429]:=

```
Total[threeBodyApseVariations]
Total[fourBodyApseVariations]
```

Out[429]=

2.18252×10^{7} m

Out[430]=

2.71243×10^{7} m

Now we will calculate the ratio relative frequency of the three body model to the four body model. If the ratio is near 1 this means that the three-body variations occur at nearly the same frequency as the four-body variations and if the ratio is much less than 1, this means that there are significantly less three-body variations than four-body variations.

In[431]:=

```
Total[threeBodyApseVariations] / Total[fourBodyApseVariations]
```

Out[431]=

0.804638

Clearly since 0.804638 is near 1, we can say that the four-body variations occur at around the same frequency as the three-body variations. We will also compare the sparseness of errors of the simulations.

In[432]:=

```
StandardDeviation[threeBodyApseVariations]
StandardDeviation[fourBodyApseVariations]
```

Out[432]=

446 597. m

Out[433]=

330 906. m

In[434]:=

```
1 - StandardDeviation[fourBodyApseVariations] / StandardDeviation[threeBodyApseVariations]
```

Out[434]=

0.259052

Variation and increased error is cumulative, so makes sense for errors to increase as time progresses. Expectedly, the four-body simulations indeed seems to have less errors than the three-body simulations. In the earlier apsis differences, the error seems to be does not explode as much and are more predictable in the four-body than the three-body. However, as the model progresses to be decade-long, the errors of the four-body and three-body with the real data both seem quite large, and the three-body model is actually 19.5% better in total in the 10 years. However, the data in the four-body is still less random and the error is more predictable overall, as shown by a 25.9% smaller standard deviation.

We suspect that Jupiter indeed plays a large role in a better model prediction, but there seems to be other factors affecting the four-body as well. This could perhaps be because we only looked at differences of distances to Earth so far.

Apsis Line

Apsis Distances

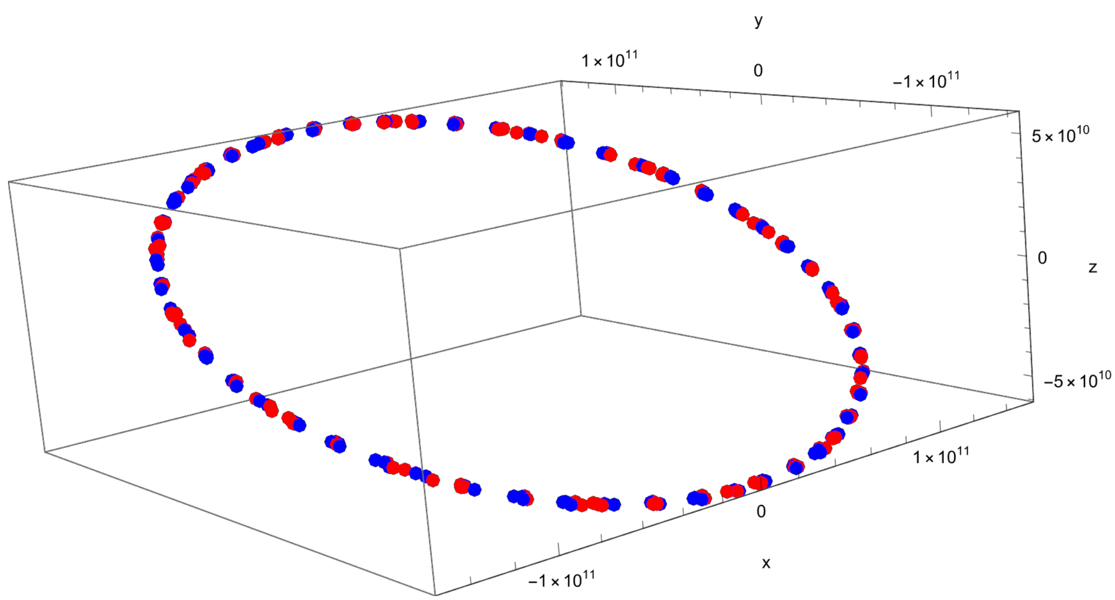
Another data point worth analyzing is the length of the apsis line, which in this project, we define it as the line directly connecting the apogee to the next perigee in space. In reality, though, this may not actually define the apsis lines, but this data is still worth comparing.

Visuals for the apsis in ListPointPlot3D. The second plot below shows the first 8 apsis lines.

In[435]:=

```
Show[
 ListPointPlot3D[moonPositionsICRS[[peaks[[All, 1]]]], PlotStyle → Blue],
 ListPointPlot3D[moonPositionsICRS[[valleys[[All, 1]]]], PlotStyle → Red],
 AxesLabel → {"x", "y", "z"}, Boxed → True, ImageSize → Large, Range → Full
]
```

Out[435]=



The above figure is a bit cluttered so we will scatter it so it is easier to see what is going on.

In[436]:=

```
scatterPlot8 = Show[
   ListPointPlot3D[moonPositionsICRS[[peaks[[All, 1]]]][[1 ;; 8]],
    PlotStyle → {Blue, PointSize[0.01]}],
   ListPointPlot3D[moonPositionsICRS[[valleys[[All, 1]]]][[1 ;; 8]],
    PlotStyle → {Red, PointSize[0.01]}],
   ListPointPlot3D[{initialSunPositionICRS1}, PlotStyle → Yellow], PlotRange → Full,
   AxesLabel → {"x", "y", "z"}, ImageSize → Large, Range → Full
  ];
```

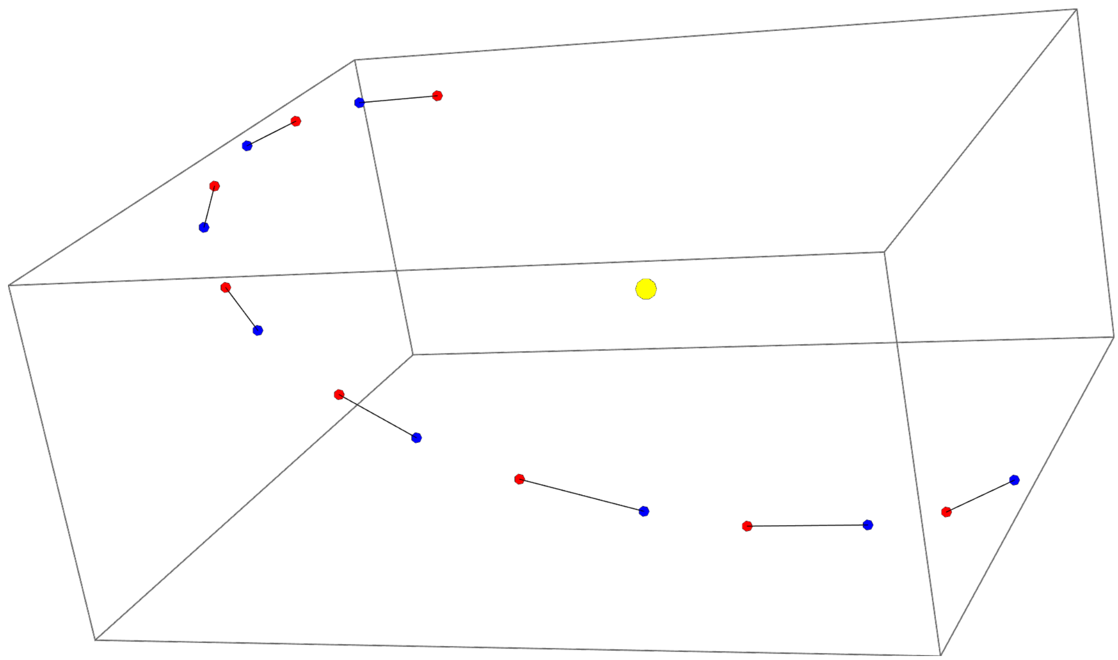
In[437]:=

```
justLines = Graphics3D[
   Line /@ Transpose[{QuantityMagnitude /@ moonPositionsICRS[[peaks⟦All, 1⟧]]⟦1 ;; 8⟧,
      QuantityMagnitude /@ moonPositionsICRS[[valleys⟦All, 1⟧]]⟦1 ;; 8⟧}]];
```

In[438]:=

```
Show[justLines, scatterPlot8, AxesLabel → {"x", "y", "z"}, ImageSize → Large]
```

Out[438]=



In[439]:=

```
apseLines = Drop[Transpose[{moonPositionsICRS[[peaks⟦All, 1⟧]],
     moonPositionsICRS[[valleys⟦All, 1⟧]]}], -1];
Iconize[apseLines]
```

Out[440]=

```
{⋯} +
```

Transpose the two matrices so that it is a list of a pair of points. Drop the last value (since it was erroneously added).

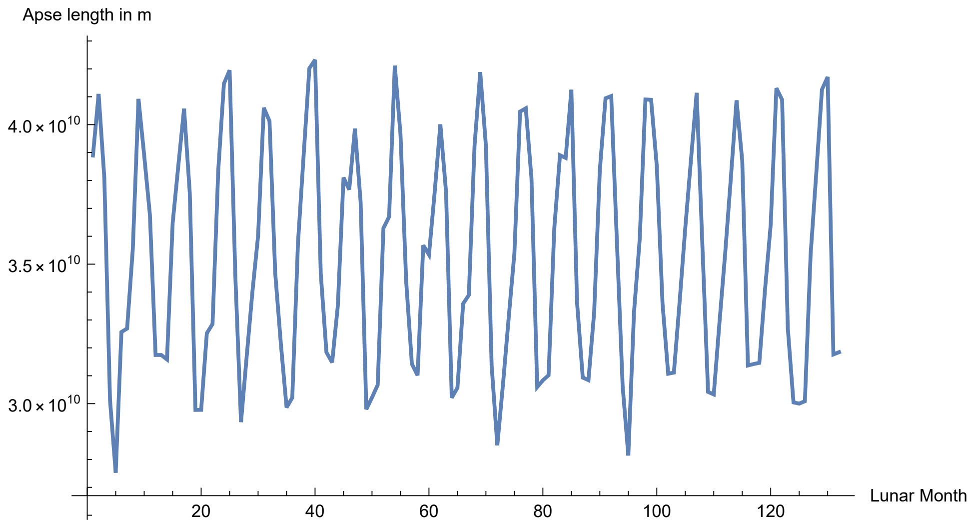
Out[∘]=

```
{⋯} +
```

In[441]:=

```
apseDistances =
  Table[EuclideanDistance[apseLines⟦i⟧⟦1⟧, apseLines⟦i⟧⟦2⟧], {i, Length[apseLines]}];
ListPlot[apseDistances, Joined → True,
 AxesLabel → {"Lunar Month", "Apse length in m"}, ImageSize → 500]
```

Out[442]=



Because of the Sun's perturbation and related factors, the length of the apsis line also fluctuates, as the perigees and apogees do too. As a result, it is also behaves in a harmonic motion.

Below, we repeat this with our 3-body simulation:

In[443]:=

```
threeBodyApseLines = Drop[Transpose[{moonSimPos3BodyICRS⟦threeBodyPeaks⟦All, 1⟧⟧,
     moonSimPos3BodyICRS[[threeBodyValleys⟦All, 1⟧]]}], -1];
Iconize[threeBodyApseLines]
```

Out[444]=

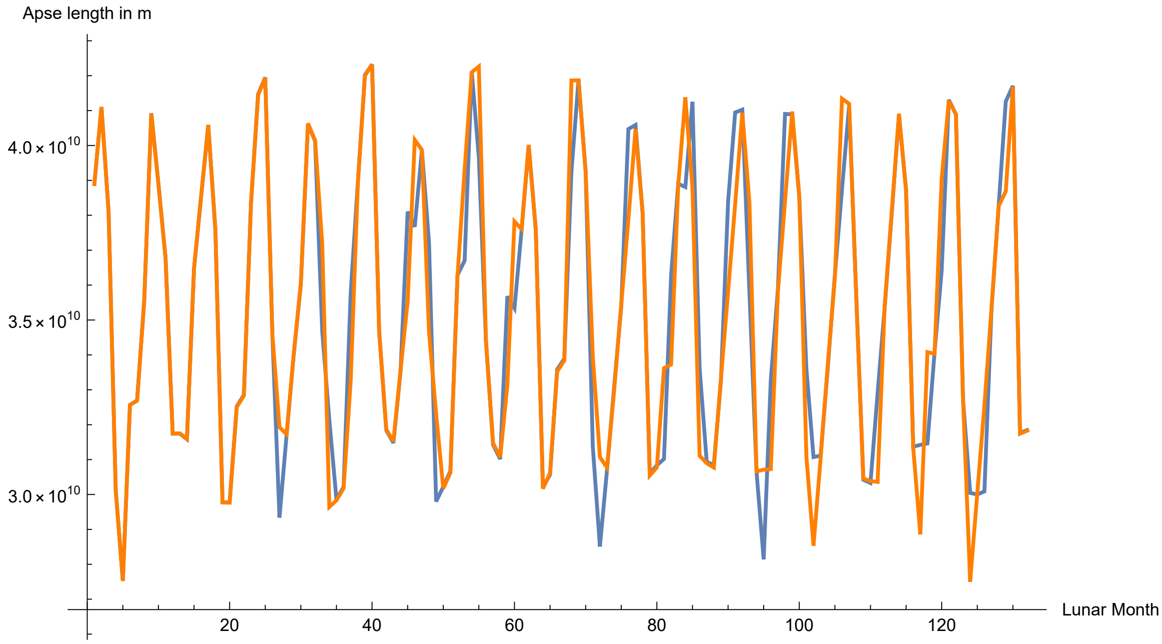
```
{⋯} +
```

In[445]:=

```
threeBodyApseDistances =
  Table[EuclideanDistance[threeBodyApseLines⟦i⟧⟦1⟧, threeBodyApseLines⟦i⟧⟦2⟧],
   {i, Length[threeBodyApseLines]}];
```

In[446]:=
```
Show[{ListPlot[apseDistances, Joined → True],
  ListPlot[threeBodyApseDistances, Joined → True, PlotStyle → Orange]},
 AxesLabel → {"Lunar Month", "Apse length in m"}, ImageSize → 600]
```

Out[446]=



The simulations pretty accurately depict the apsis line lengths when lengths are not at its maxima or minima, where most of the errors occur. The data, interestingly, has its greatest errors at the minima of apse lengths, where the model's distance either does not dip enough or dips too much. Perhaps the three-body model is not accounting for some factors that are amplified when the Moon's extrema orbit is especially close.

Apse Angles Shift

Lastly, another data point worth considering is the angle change between the apsis lines. If we fix Earth as the center, how would the apsis line change over time? This problem was explored by many great scientists in the 17th and 18th century. For example, Newton solved this problem analytically but made an error in his calculations and was off by a factor. We will try to solve this problem computationally!

In[447]:=
```
moonAngle[d_] :=
 AstroPosition[ Moon PLANETARY MOON , {"Ecliptic", "Date" → d}, "Spherical"]["Longitude"]
```

In the ecliptic coordinate system with the Earth in the center, the longitude of the Moon would be its angle on the ecliptic plane.

Using this, we can find how the angle between apse lines change by plotting how the angle between the apogees and the Earth changes in the ecliptic plane.

In[448]:=

```
apos = ⊟ Sat 1 Jan 2000 00:00:00 UTC ✓ + Quantity[#, "Days"] & /@ peaks[[All, 1]];
```

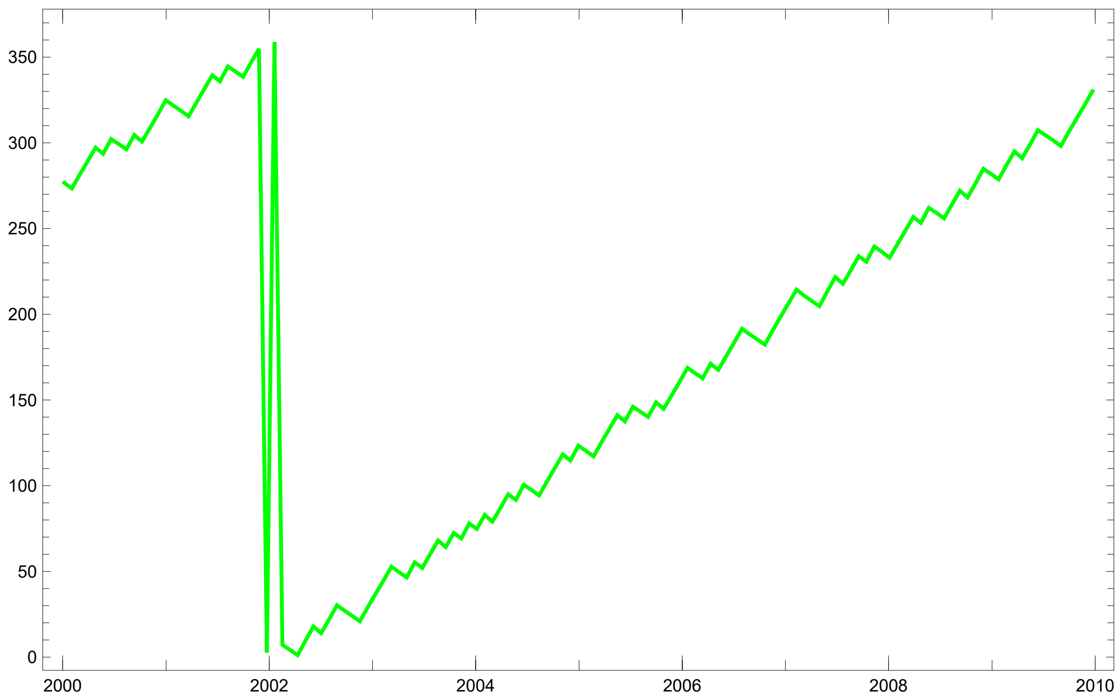
In[449]:=

```
aposMovementPlot;
```

In[450]:=

```
Module[{angles, tab},
 angles = Table[moonAngle[d], {d, apos}];
 tab = Table[{apos[[n]], angles[[n]]}, {n, Length[apos]}];
 aposMovementPlot =
  DateListPlot[tab, Joined → True, ImageSize → Large, PlotStyle → {Green}]]
```

Out[450]=



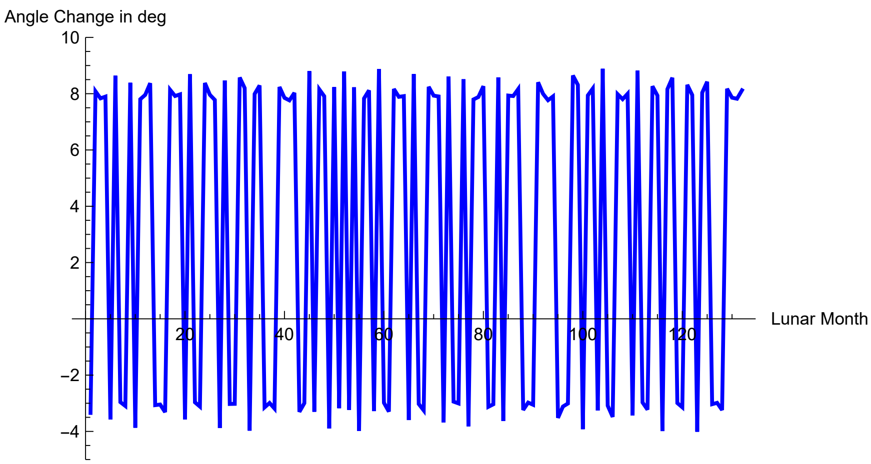
From the graph above, there is a relatively linearly increasing pattern for the angle of the Moon's orbit relative to Earth. It makes sense that there is an apsidal precession of the elliptical orbit of the moon, due to the many factors stemming from the sun's perturbation. In particular, as discussed by Nesterenko [arXiv:2310.05584], the Sun's perturbation can result in corresponding perturbations projected on the central force, f(a). As demonstrated by the formula, perturbation causes the ellipse to have apsidal precession (gradual rotation of the apse line): $\delta_{aps} = 2\pi\left(\sqrt{\frac{f(a)}{3f(a)+f'(a)}}-1\right)$. An estimate for rotation per lunar month can then be calculated.

Below, we can see how much the angle changes by with successive apses. There are quite huge variations, which also almost resembles a harmonic motion in a way. Nevertheless, the general pattern is an increase in the angle by 3.13 degrees per lunar cycle. Our real-world data's calculation is very close to Nesterenko’s calculation of an average 3.02 degrees per lunar month, and like Nesterenko, our data provides further evidence against Newton’s erroneous calculation of 1.52 degrees in “Principia.”

In[451]:=

```
aposAngleDif = If[# > 180, # - 360, If[# < -180, # + 360, #]] & /@
   Table[moonAngle[apos[[i + 1]]] - moonAngle[apos[[i]]], {i, 1, Length[apos] - 1}];
plotAposAngleDif = ListPlot[aposAngleDif, Joined → True, PlotRange → {-5, 10},
  PlotStyle → Blue, AxesLabel → {"Lunar Month", "Angle Change in deg"}, ImageSize → 450]
Quantity[Mean[aposAngleDif], "Degrees"]
```

Out[452]=



Out[453]=

3.13209 °

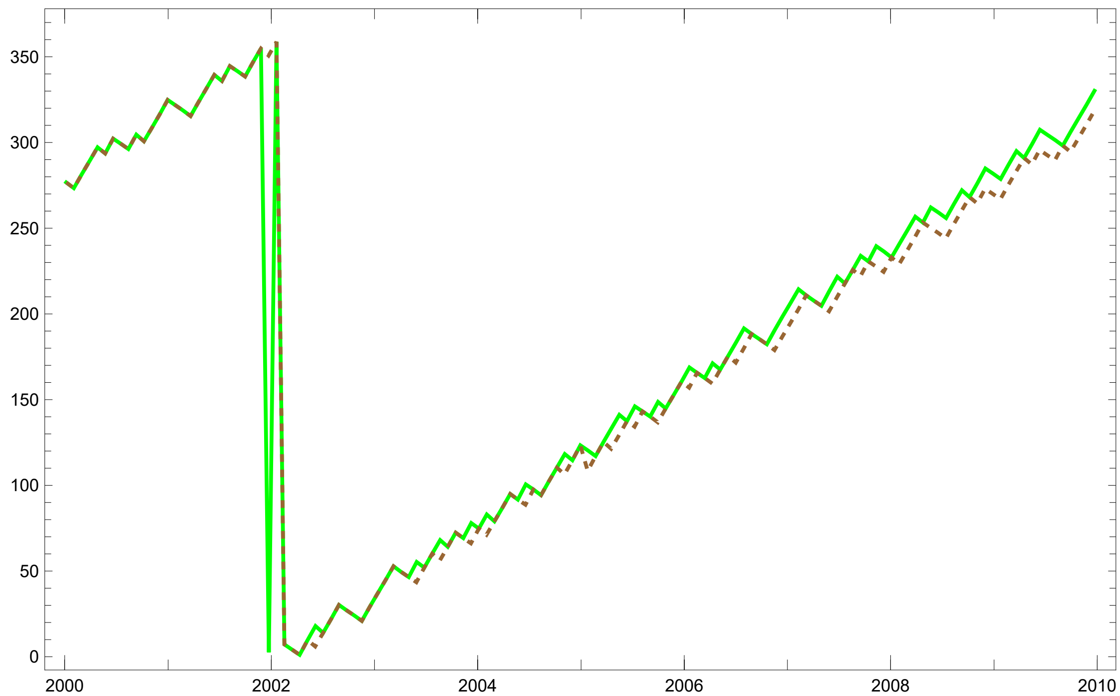
Now, let’s do something similar with the 3-body.

In[454]:=

```
threeBodyApos =
  Sat 1 Jan 2000 00:00:00 UTC + Quantity[#, "Days"] & /@ threeBodyPeaks[[All, 1]];

Module[{angles, tab},
 angles = Table[moonAngle[d], {d, threeBodyApos}];
 tab = Table[{threeBodyApos[[n]], angles[[n]]}, {n, Length[threeBodyApos]}];
 Show[aposMovementPlot,
  DateListPlot[tab, Joined → True, ImageSize → Large, PlotStyle → {Brown, Dashed}]]]
```

Out[455]=



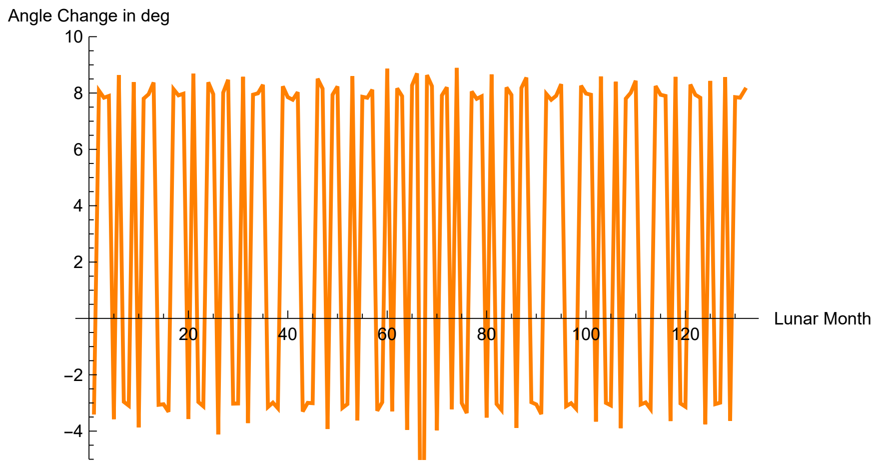
From this plot, we see that the three-body model follows the general pattern of the Moon’s apsidal precession really well. However, the three-body’s angle change seems to simplify some ups and downs found in the actual data. This is particularly evident after 2006, where the three-body’s prediction is a bit lower and tends to miss some dips. This suggests that perhaps imbalanced perturbations from other planetary objects can may not cause much effect in the patterns overall, but may result in nuanced changes and oscillations, which would accumulate some error in the three-body model.

Simply looking at the angle change per lunar month, we see a very similar shape with the actual angle change rate.

In[456]:=

```
threeBodyAposAngleDif =
  If[# > 180, # - 360, If[# < -180, # + 360, #]] & /@ Table[moonAngle[threeBodyApos[[i + 1]]] -
     moonAngle[threeBodyApos[[i]]], {i, 1, Length[threeBodyApos] - 1}];
threeBodyPlotAposAngleDif = ListPlot[threeBodyAposAngleDif,
  Joined → True, PlotRange → {-5, 10}, PlotStyle → Orange,
  AxesLabel → {"Lunar Month", "Angle Change in deg"}, ImageSize → 450]
Quantity[Mean[threeBodyAposAngleDif], "Degrees"]
```

Out[457]=



Out[458]=

3.04274 °

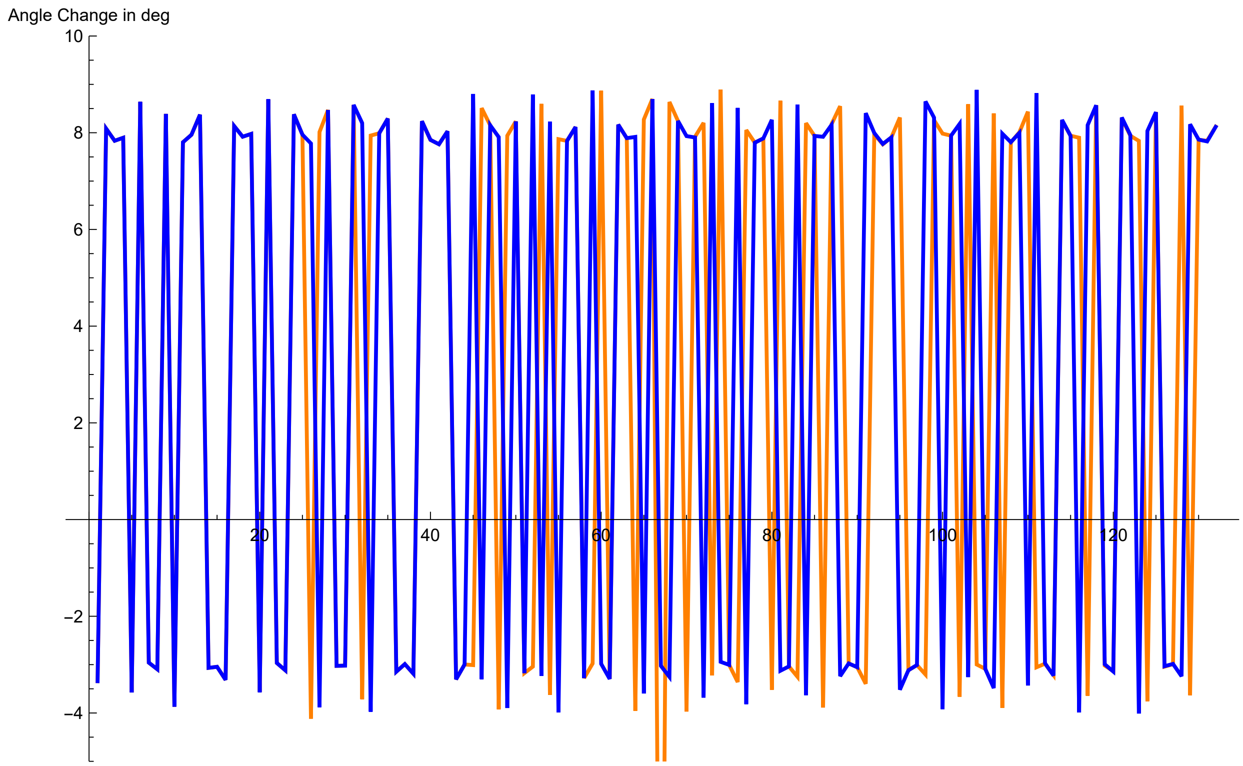
Our result of 3.04274 degrees is closer to Nesterenko's prediction of 3.02 than the real data, as he used a similar three-body system.

Below, we overlaid the orange graph (3-Body apogee angle rate of change) on the blue graph (actual apogee angle rate of change):

In[459]:=

```
Show[threeBodyPlotAposAngleDif, plotAposAngleDif,
 ImageSize → 700, AxesLabel → {"Lunar Month", "Angle Change in deg"}]
```

Out[459]=



The angle error of the change in apse line's angle per lunar month does not seem to increase too much in the 3-body, as the graphs maintain rather close together and have very similar averages. In fact, at many instances, the model seems to correct itself to be closer to actual data after straying off its pattern (such as when the 3-body plot is quite a bit to the left in the 121st month, but quickly returns to be nearly identical on the 125th month). Although this may be coincidental, it further suggests that the 3-Body model performs very well in predicting the overall pattern of the Moon's apse line progression, even if it may miss some small nuances that may get corrected or compensated over time.

Conclusion

In this extension, we continued to look at how well the 3-body, 4-body simulation results predict the orbit of the Moon, Earth, and Sun using data from our last paper and AstroPosition through the lens of the line of apses. We analyzed the apogees and perigees individually, and also together (differences in the distances). We also analyzed and compared the model with actual data with several other apse perspectives: apse length by connecting the apogee and perigee and the apsidal precession angular change.

Our findings also suggest that increasing the number of bodies would make the errors in the model more stable and in general more accurately predict the actual motion of the apogees and perigees. These benefits of the four-body is limited, however, as within 10 years the data points have also diverged and the variance can become greater than the three-body's.
In terms of the apse line, the three-body model already predicts the general pattern of lengths and angle of rotation in the ecliptic coordinate system very well. It may miss small nuances and variations in the real data, possibly due to other factors and phenomena we did not account for. Nevertheless, by considering the sun's motion and perturbation in our model, general error of our model remains minimal and may even self-correct.

Further research could include more bodies, different calculations, and analyses of how the non-spherical shape of the planets may affect apse patterns, if significantly at all.

References

1. NASA. (n.d.). Eclipses and the Moon's orbit. NASA. https://eclipse.gsfc.nasa.gov/SEhelp/moonorbit.html

2. NASA's Scientific Visualization Studio. (2023, December 7). Moon's precession and the 18.6-year cycle. NASA. https://svs.gsfc.nasa.gov/5326/

3. Nesterenko, V. V. (2023). Simple calculation of the Moon apsides motion. arXiv. https://arxiv.org/pdf/2310.05584

4. Wolfram, Stephen. "When Exactly Will the Eclipse Happen? A Multimillennium Tale of Computation." Stephen Wolfram Writings, 27 Mar. 2024, https://writings.stephenwolfram.com/2024/03/when-exactly-will-the-eclipse-happen-a-multimillennium-tale-of-computation/. Accessed 29 Jan. 2025.