Analysis of Numerical Methods for Planetary Orbit Prediction

Varun Kesharaju and Atharva Navnath Nimbalkar
(https://github.com/kvv1618/NumericalComputations)

1. Abstract:

This project investigates Venus's orbital prediction using three approaches: Kepler's equations, regression, and interpolation. Keplerian mechanics were employed to derive orbital elements, addressing computational challenges with high-precision arithmetic. Regression models, enhanced with trigonometric features, captured the periodic nature of Venus's motion, yielding accurate predictions. Interpolation techniques provided efficient short-term predictions of Venus's position, validated through error metrics and visual comparisons. These methods collectively demonstrate the trade-offs in accuracy and computational complexity, offering insights for celestial orbit modeling.

2. Introduction:

Predicting celestial orbits has been a cornerstone of astronomy, enabling advancements in space exploration, satellite positioning, and planetary science. Venus, as Earth's closest planetary neighbor, offers a unique case study for orbit prediction due to its relatively short orbital period and near-circular trajectory. This project aims to explore and evaluate various numerical methods for modeling Venus's spatial orbit, highlighting the trade-offs between precision, computational efficiency, and adaptability to large datasets.

The study is divided into three distinct approaches: Kepler's equations, regression techniques, and interpolation methods. Each approach serves a unique purpose, reflecting the diversity of tools available for orbit prediction.

1. Kepler's Equations:
   Grounded in celestial mechanics, Kepler's laws provide a deterministic framework for orbit prediction. Using high-precision arithmetic, we derived key orbital elements such as semi-major axis, eccentricity, inclination, and argument of periapsis. These elements were then used to calculate Venus's position at future points through transformations from mean anomaly to true anomaly and finally into Cartesian coordinates. The approach highlights challenges such as division instability and error propagation, particularly when handling extremely small eccentricities or large numerical values.

2. Regression Techniques:
   Modern machine learning offers powerful tools for capturing complex, non-linear patterns in data. Regression models, including linear, polynomial, and advanced ensemble methods like XGBoost and Random Forest, were applied to the time series of Venus's orbital data. Trigonometric features were incorporated to

account for the periodic nature of planetary motion, significantly improving model performance. By splitting the data into training and test sets, we evaluated each model using metrics such as RMSE, R^2, and cross-validation, ensuring robust error analysis.

3. Interpolation Methods:
   Interpolation provides a flexible, computationally efficient alternative for orbit prediction over short timescales. Using cubic and quadratic spline interpolation, we generated continuous approximations of Venus's position based on observed data. These models were tested for their ability to extrapolate over one orbital period and were compared against ground truth data using both error metrics and graphical visualizations.

Through these approaches, we address key questions about the trade-offs between traditional physics-based methods and data-driven or numerical techniques. The findings underscore the importance of methodological diversity in planetary science, particularly in contexts where computational resources and data availability vary.

3. Background:

Venus is said to have a nearly-circular orbit. Therefore the Sun can be considered as the origin, making Venus' orbit plotting easier. NASA's JPL (Jet Propulsion Laboratory) has an Horizons System platform, where historic data for Venus orbit can be downloaded with the Sun being at (0,0,0) coordinate center.

Data Collection:

1. The "App" tab on this webpage prompts for a few settings input in order to retrieve and generate Ephemeris.
   1.1. Ephemeris type: Tables of Cartesian vectors (e.g. xyz coordinates in a specific reference frame) are available using the "Vector Table" type.
   1.2. Coordinate Center: 0°E, 0°N, 0 km @10 (Sun [Sol]) (observer table as seen from the Sun)
   1.3. Time Specification: We have collected 30 Earth years of data, from 1992-Sep-30 to 2022-Sep-30, with 1 Earth day as the timestep.

Data preprocessing:

1. Relevant information such as timestamp, cartesian coordinates and velocity components are extracted into a csv from the downloaded text file.
   1.1. ETL is performed by keeping all the numerical values as strings because we want to preserve the precision.

Understanding the orbital elements:

Semi-Major Axis: The semi-major axis represents the longest radius of an ellipse, measured from its center to the farthest edge along the major axis.

Eccentricity: A dimensionless parameter that quantifies the deviation of an orbital path from a perfect circle. Eccentricity values range between 0 and 1, where:

- e=0: Indicates a perfectly circular orbit.
- e=1: Represents a parabolic trajectory.

Inclination: The angle formed between the orbital plane and the reference plane, commonly the ecliptic plane, which corresponds to Earth's orbital path around the Sun.

Longitude of the Ascending Node: The angular measurement from a fixed reference direction (e.g., the vernal equinox) to the ascending node, where the orbit transitions from below to above the reference plane.

Argument of Perihelion: The angular distance between the ascending node and the perihelion point, where the orbiting body is closest to the central focus (e.g., the Sun). An argument of perihelion of 0° implies the perihelion coincides with the ascending node, while 90° indicates the perihelion occurs at the point where the orbiting body is furthest above the reference plane.

4. Methodology:

The goal of our project was implementing different numerical methods that will help us analyze the stability and accuracy when handling huge numerical values. Once we had sufficient background domain knowledge, we carried out different methods to predict Venus' spatial orbit. For each method that is carried through, we have done error analysis using metrics such as RMSE, R2, and graphical visualisations.

1. Kepler's Equations:

Firstly we had to read the data not as a simple dataframe from a csv at once, but line-by-line as numpy arrays of float128 data-type. Pandas' traditional *read_csv()* method will load data as float64 values. Numpy's float128 data-type provides 80 bits to store a decimal value, providing the ability to store values in the interval, *[-1.189731495357231765e+4932, 1.189731495357231765e+4932].*

	Date	Position_x	Position_y	Position_z	Velocity_x	Velocity_y	Velocity_z
0	1992-09-30	-2.353658e+07	-1.056952e+08	-42094.061217	32.644443	-9.032771	-1.945275
1	1992-10-01	-2.069097e+07	-1.064476e+08	-212555.484600	33.233729	-8.366555	-2.001262
2	1992-10-02	-1.779289e+07	-1.071377e+08	-387975.673075	33.854380	-7.588821	-2.059448
3	1992-10-03	-1.484083e+07	-1.077554e+08	-568383.287081	34.478586	-6.689384	-2.116079
4	1992-10-04	-1.183574e+07	-1.082900e+08	-753480.878332	35.076722	-5.665242	-2.167398

Figure 1: Raw input dataframe

Angular momentum, which is the cross product of position vector with velocity vector, is the base for deriving all the orbital elements.

1. Semi-major axis:

Formula : $a = 1/((2/r) - (v^2/GM))$

(where GM is taken as *1.32712440018e11* km^3/sec^2)

The Value obtained using this formula is *109444168.34983853* kms, while the true value of the semi-major axis of Venus is *108200000* kms.

This gives us a relative error of *1.1498783270226633111*% ,which is calculated as $((calculated\ value\ -\ true\ value)\ /\ true\ value)\ *\ 100.$

2. Eccentricity:

Formula : $e = (Aphelion - Perihelion)/(Aphelion + Perihelion)$

There was a huge difference between the calculated eccentricity value and the true eccentricity. Upon careful observation, we have found that this is because of instability of division operation in computers.

Consider the values,

$a = 0.728,\ b = 0.718,\ a' = 0.73288130374203175332,\ b' = 0.71376480875589953801.$

If the relative error between a and a', b and b' is relatively less, i.e $((a' - a)/a) * 100 = 0.6\%$, $((b - b')/b) * 100 = 0.5\%$. Yet when division is performed, the relative error between the results $(a - b)/(a + b)$, $(a' - b')/(a' + b')$ is huge (around *98%*). As a matter of fact, $a\ and\ b$ are the true Aphelion and Perihelion numbers of Venus, whereas $a'\ and\ b'$ are the calculated values of the same (All in Astronomical units).

To minimise the error in eccentricity calculation, we have tried to utilise the GNU Multiple Precision Arithmetic Library(GMP). We have defined every value as a gmp variable of 128 bits length and performed all calculations upon those variables using methods defined in GMP, yet the result was the same as initial, which is *0.013214354790007146316,* its true value being *0.0067.*

3. Mean inclination:

Formula :

$i = cos^{-1}(z\ component\ of\ angular\ momentum/\ magnitude\ of\ angular\ momentum)$

The Value obtained using this formula is *3.3940019733655222 degrees*, while the true value of inclination of Venus is *3.39* degrees.

This gives us a relative error of *0.11805231166732356996*%.

4. Mean Longitude of ascending node:

Formula :

$\Omega = tan^{-1}(x\ component\ of\ angular\ momentum/(-\ 1\ *\ y\ component\ of\ angular\ momentum))$

The Value obtained using this formula is *76.65871657844771 degrees*, while the true value is *76.785* degrees.

This gives us a relative error of *0.16446366028819717857*%.

5. Mean Argument of periapsis:

Formula :

$\omega = cos^{-1}(n'.e')$, where n' is the unit vector of node vector and e' is the unit vector of eccentricity.

Node vector is calculated as the cross product of the unit vector normal to the reference plane, usually taken as (0,0,1) and angular momentum vector.
The Value obtained using this formula is *88.29658383476765 degrees*, while the true value is *54.78* degrees. Such an error is because of the propagating error in eccentricity.
This gives us a relative error of *61.18397925295297631%*.

Once we had necessary orbital elements, we had predicted the position of Venus through the next year.
It has to be noted that Venus takes 225 Earth days to complete one revolution around the Sun. Hence, we took an array of 0 to 225 days, and calculated the mean anomaly at each point, which is the time array of length 225 multiplied by $2 * \pi/225$.

Next step would be to calculate eccentric anomaly using iterative methods like Newton Raphson's, with initial guess being equal to mean anomaly.

```
E = mean_anomaly
E_nxt = E - (E - e*np.sin(E) - mean_anomaly)/(1 - e*np.cos(E))
i = 0
while not(np.all(np.abs(E - E_nxt) < 0.5*(10**-7))):
    i+=1
    E = E_nxt
    E_nxt = E - (E - e*np.sin(E) - mean_anomaly)/(1 - e*np.cos(E))
E = E_nxt
```

Figure 2. Iterative code block used to calculate eccentric anomaly.

True anomaly is derived from eccentric anomaly using the relation;
$\boldsymbol{\nu} = 2 * tan^{-1}(\sqrt{(1 + e)/(1 - e)} * \ tan(E/2))$; where e is the eccentricity value.
This equation would result true anomaly, but the output has to be mapped from tan inverse range $[-\pi/2, \pi/2]$ to $[0, \pi]$ radians.
Calculating radial distance from Sun and thereafter cartesian coordinates would follow a series of transformations using a rotation matrix applied on perifocal coordinates calculated as;
$x = radius * cos(\boldsymbol{\nu}),\ y = radius * sin(\boldsymbol{\nu}),\ z = 0$ where
$radius = a * (1 - e^2)/(1 + e * cos(\boldsymbol{\nu}))$, a being the semi-major axis.

Visual representation of final results;

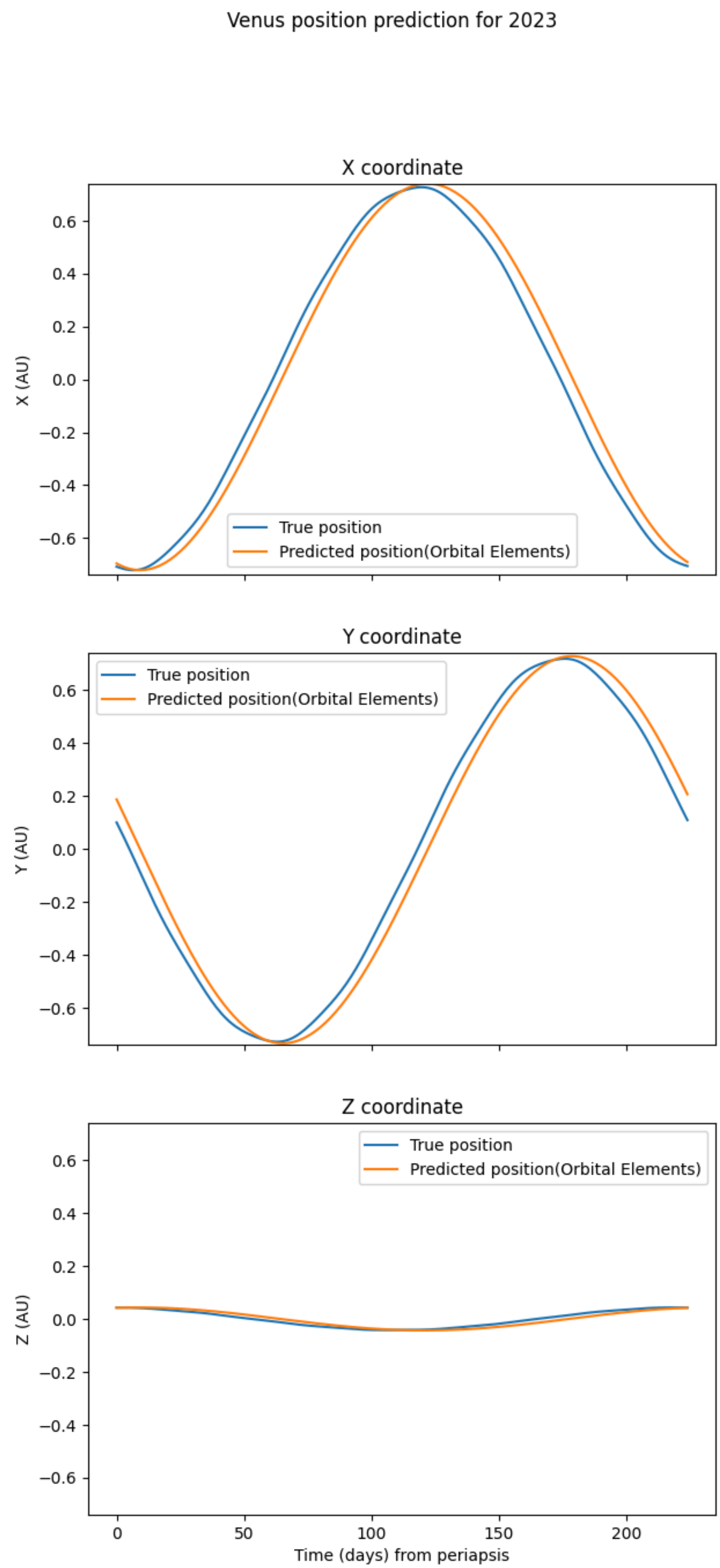

Venus position prediction for 2023
X coordinate
X (AU)
True position
Predicted position(Orbital Elements)
Y coordinate
Y (AU)
True position
Predicted position(Orbital Elements)
Z coordinate
Z (AU)
True position
Predicted position(Orbital Elements)
Time (days) from periapsis

Figure 3. Predicted and Actual individual cartesian values overlapped over the shared X axis.

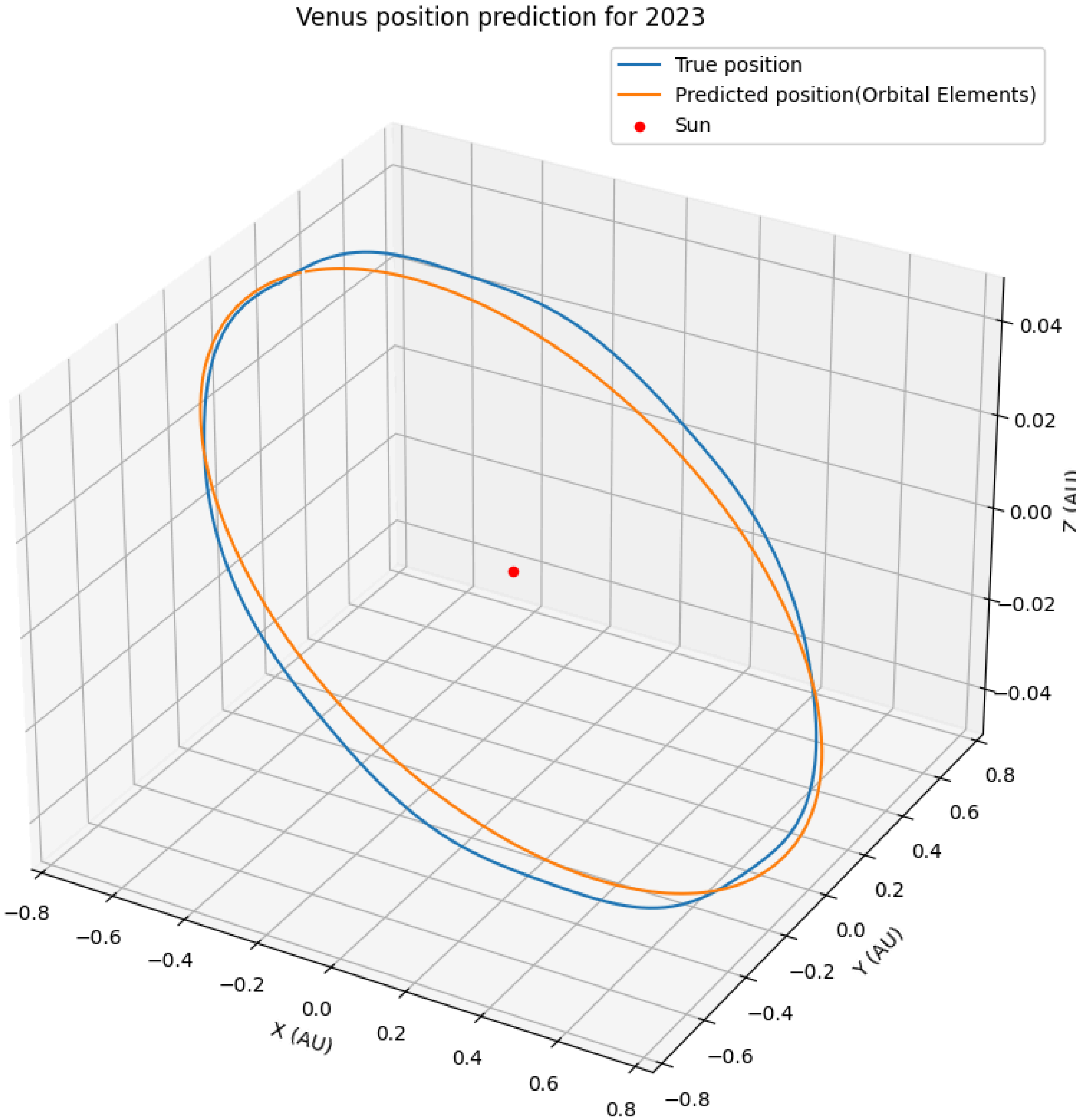


Figure 4. 3D plot of predicted VS actual orbit of Venus.

Propagation error can be clearly seen in the final error evaluation of the predicted values.

```
Unset
Mean Squared Error: 0.00810751031130828 AU^2
Root Mean Squared Error: 0.09004171428459301 AU
Mean Absolute Error: 0.12152170804632072 AU
```

```
Final Error Percentage: 15.12%
```

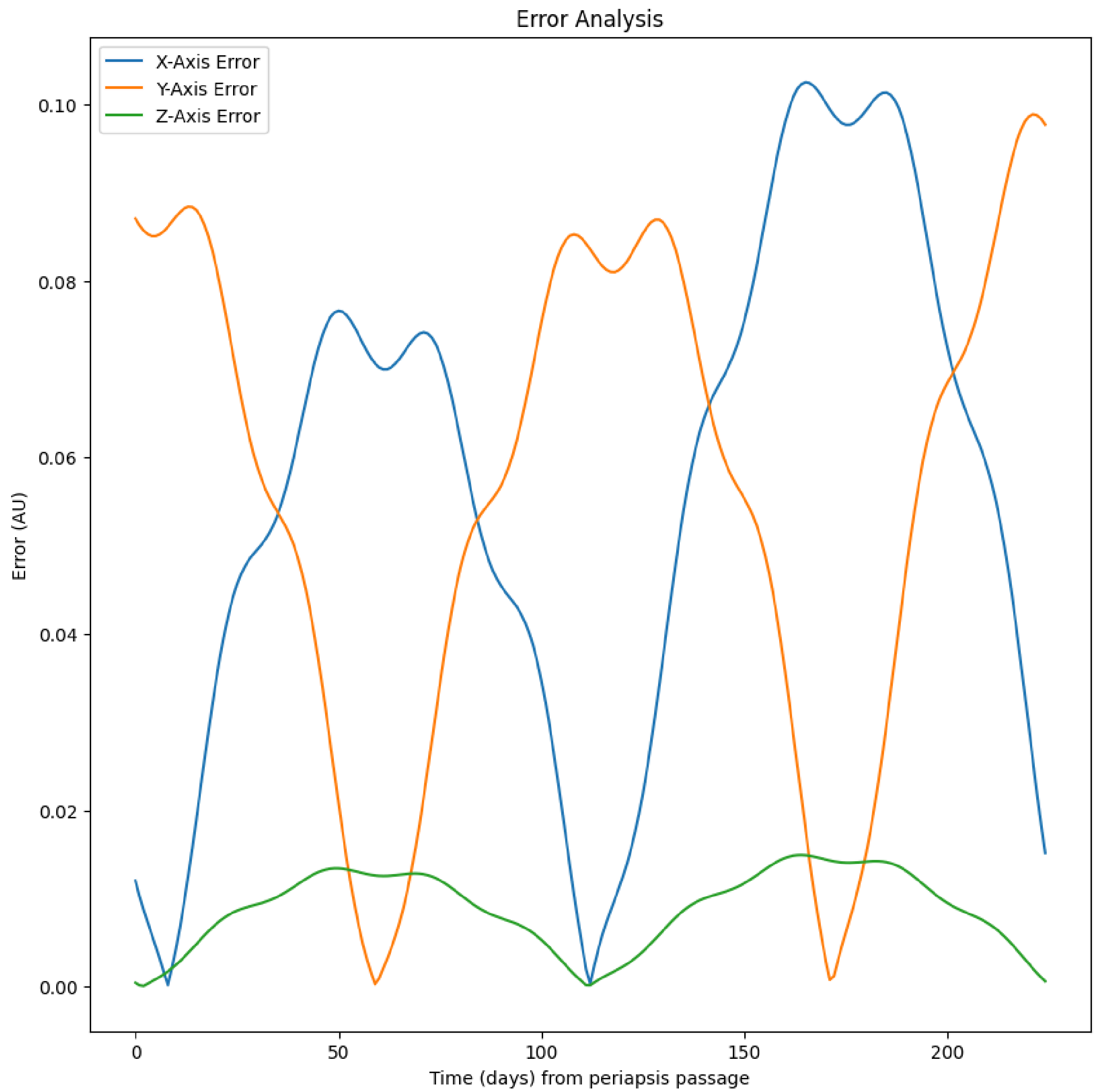


Figure 5. Plot of individual axis error values in Astronomical units.

2. Regression

This study applies numerical methods and machine learning techniques to analyze and predict Venus's spatial orbit. The following steps outline the methodology, emphasizing the systematic actions undertaken to achieve accurate and computationally efficient results.

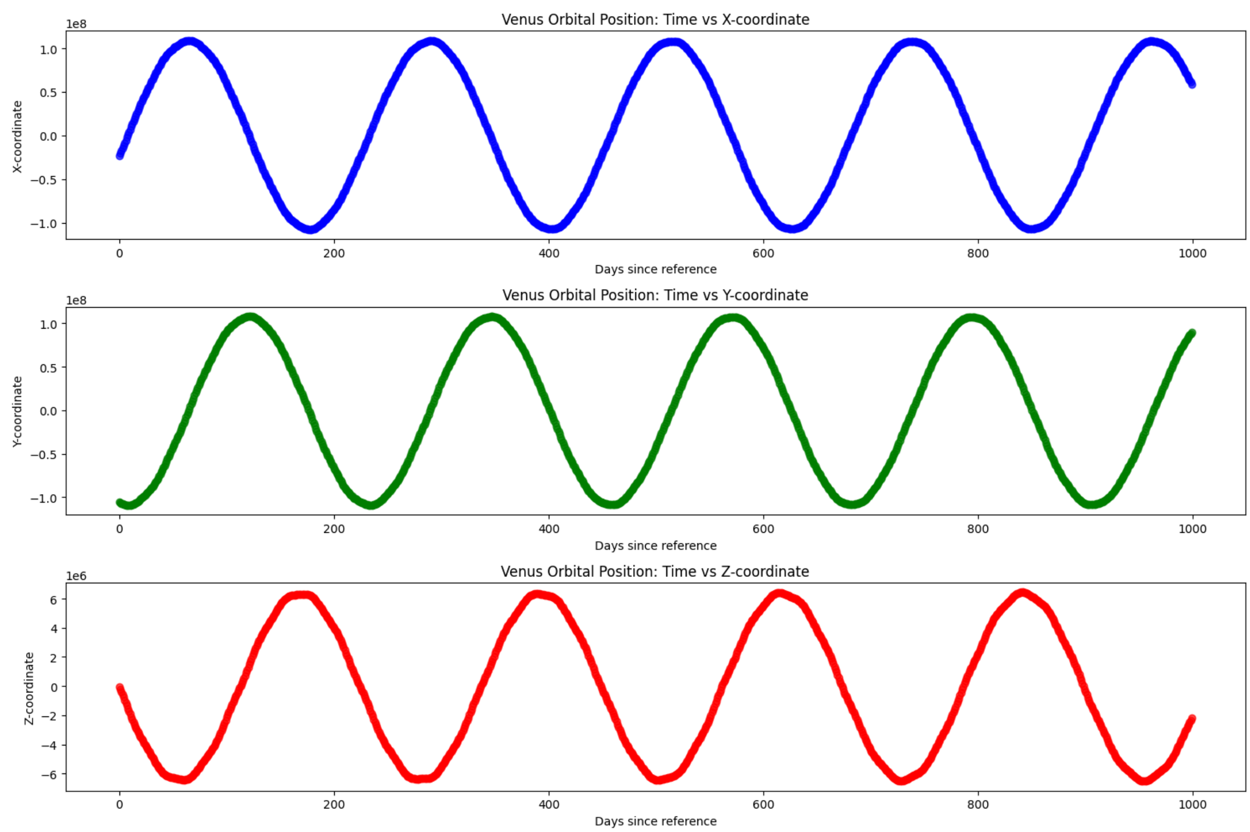
1. **Data Acquisition and Preprocessing**

- We began by acquiring a dataset that included Venus's positional (x, y, z) and velocity data spanning from September 30, 1992 to September 30, 2022.
- The data was stored in a CSV file, which we read line-by-line to ensure high precision in numerical calculations.
- We converted dates into numerical values representing days since a reference epoch to facilitate numerical computations X=date−date.min().
- The dataset was then split into feature variables (time) and target variables (spatial coordinates).:
- To ensure consistency and reproducibility, we fixed the train test split to a fixed ratio (80:20)
- The position coordinates (x, y, z) were treated as separate target variables, each requiring individual modelling.

2. **Baseline Regression Analysis**
    - We first applied linear regression to establish a baseline for further analysis.
    - To better capture non-linear relationships, we implemented polynomial regression with degrees ranging from 1 to 4.
    - Separate models were trained for each spatial coordinate (x, y, z).
    - We evaluated model performance using two metrics:
        - Root Mean Square Error (RMSE) to quantify prediction error variability.
        - R-squared (R^2) to measure the variance explained by the model.

```
Analysis for X Coordinate:
Polynomial Degree 1:
  RMSE: 76922422.07730311
  R2: -0.0003828063497322009б
Polynomial Degree 2:
  RMSE: 76912645.70820361
  R2: -0.00012853744359508923
Polynomial Degree 3:
  RMSE: 76905406.90125616
  R2: 5.9712398714384385e-05
Polynomial Degree 4:
  RMSE: 76889519.56137615
  R2: 0.0004728107989623309

Analysis for Y Coordinate:
Polynomial Degree 1:
  RMSE: 75962384.33944452
  R2: -0.0003340573638415112
Polynomial Degree 2:
  RMSE: 75981787.07494992
  R2: -0.0008451444263679608
Polynomial Degree 3:
  RMSE: 75956136.58755767
  R2: -0.000169513240314334
Polynomial Degree 4:
  RMSE: 75970396.89429224
  R2: -0.000545100087050665

Analysis for Z Coordinate:
Polynomial Degree 1:
  RMSE: 4546245.623255358
  R2: -5.5233446006885956e-05
Polynomial Degree 2:
  RMSE: 4545751.312148991
  R2: 0.00016222578310065572
Polynomial Degree 3:
  RMSE: 4544823.018616731
  R2: 0.0005705403198075221
```

3. **Visualization and Error Analysis**
    - We visualized actual data points against predicted values for each coordinate to assess model performance.
    - These visualizations highlighted trends, periodic behaviors, and systematic errors in simpler models.
    - Through this process, we identified the need for more sophisticated approaches to capture orbital dynamics effectively.
4. **Incorporation of Periodic Features**
    - Observing the periodic nature of Venus's orbit, we enhanced our features by incorporating sine and cosine transformations of the time variable.
        - sin(2π×day_of_year/365)
        - cos(2π×day_of_year/365)
    - This feature engineering allowed us to model the cyclical motion of the orbit more effectively, improving accuracy and reducing errors associated with non-linearity of the date variable.



5. **Regularized Regression Techniques**
    - We addressed overfitting in higher-degree polynomial models by applying Ridge (L2 regularization) and Lasso (L1 regularization) regression.
    - These techniques penalized model complexity, enhancing generalization capabilities.

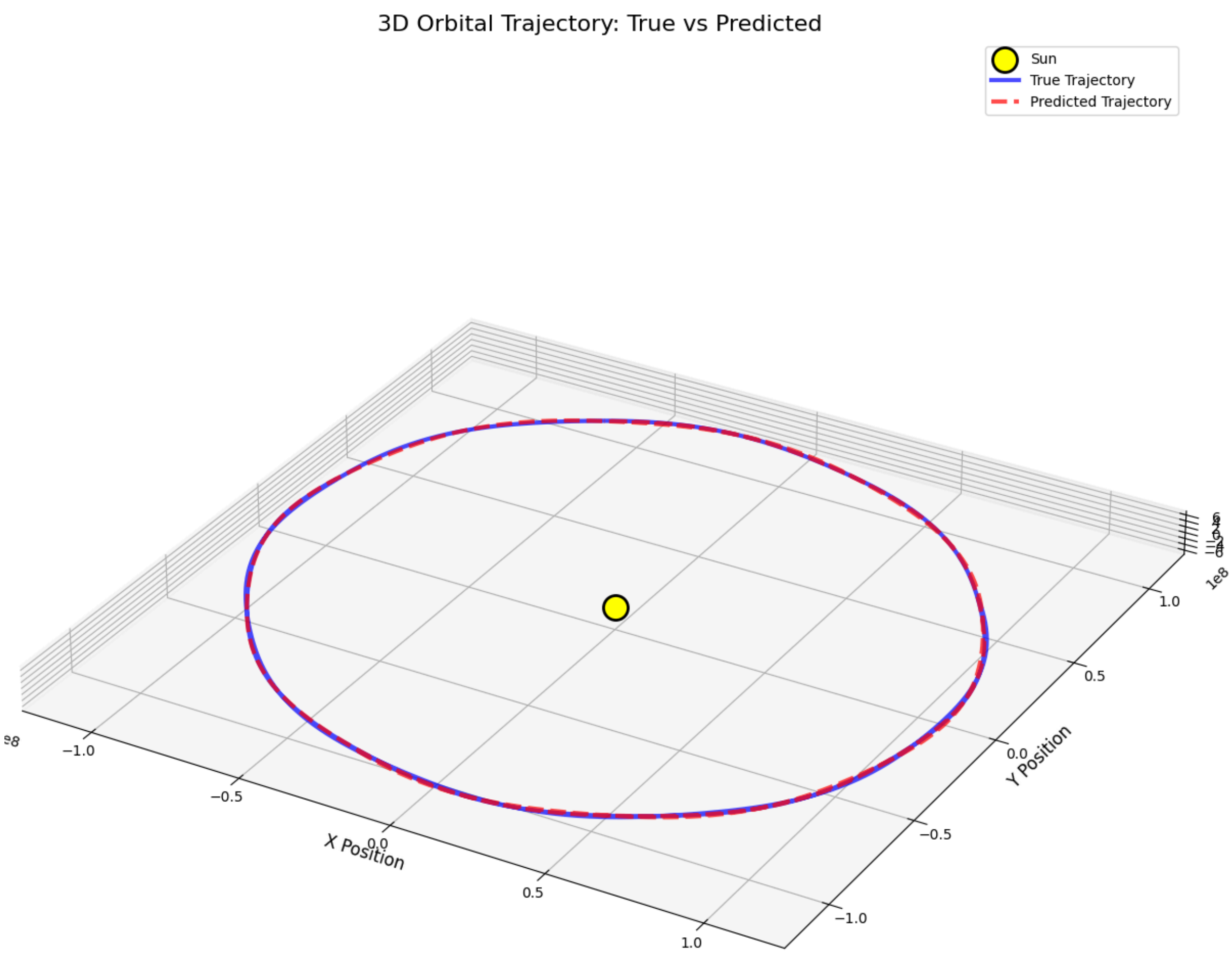
- We systematically compared their performance against unregularized polynomial models to evaluate their effectiveness in numerical predictions.

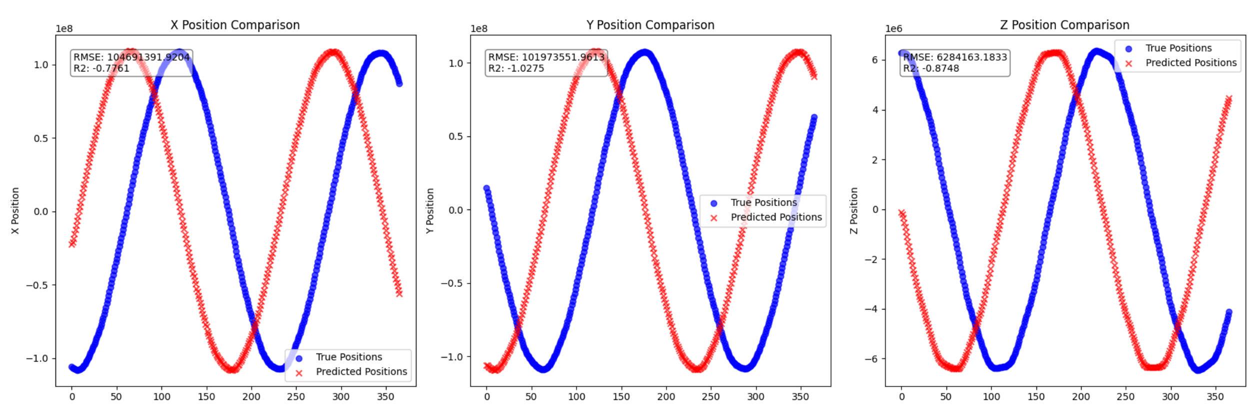
6. **Advanced Model Exploration**
    - To further enhance prediction accuracy, we employed advanced machine learning models, including Random Forest (RF) and Extreme Gradient Boosting (XGB).
    - We tested these models with and without the inclusion of velocity features to evaluate the significance of additional dynamic information.
    - Among all models tested, the Random Forest model trained without velocity features demonstrated superior performance.

```

Results without velocity features:

X Coordinate Results:

Random Forest:
  RMSE: 1374048.4986
  R2: 0.9997
  CV R2: -1.0370

XGBoost:
  RMSE: 27203996.2287
  R2: 0.8749
  CV R2: -1.0455

Y Coordinate Results:

Random Forest:
  RMSE: 1414928.4545
  R2: 0.9997
  CV R2: -1.0519

XGBoost:
  RMSE: 27561597.2035
  R2: 0.8683
  CV R2: -0.7770

Z Coordinate Results:

Random Forest:
  RMSE: 82799.5198
  R2: 0.9997
  CV R2: -1.0308

XGBoost:
  RMSE: 1633147.8549
  R2: 0.8709
  CV R2: -0.9832
```

```
Results with velocity features:

X Coordinate Results:

Random Forest:
  RMSE: 1648189.5808
  R2: 0.9995
  CV R2: -1.2218

XGBoost:
  RMSE: 14979324.7943
  R2: 0.9621
  CV R2: -1.1856

Y Coordinate Results:

Random Forest:
  RMSE: 1766931.4791
  R2: 0.9995
  CV R2: -1.1058

XGBoost:
  RMSE: 14979742.6256
  R2: 0.9611
  CV R2: -1.1275

Z Coordinate Results:

Random Forest:
  RMSE: 123869.6813
  R2: 0.9993
  CV R2: -1.1706

XGBoost:
  RMSE: 1124229.7058
  R2: 0.9388
  CV R2: -1.1690
```

7. **Prediction and Generalization**
   - Using the best-performing Random Forest model, we predicted Venus's position for a new dataset.
   - We compared predicted values against actual values through visualizations, which confirmed the model's ability to generalize and accurately capture orbital dynamics.


3D Orbital Trajectory: True vs Predicted
Sun
True Trajectory
Predicted Trajectory
X Position
Y Position
1e8
−1.0
−0.5
0.0
0.5
1.0


X Position Comparison
RMSE: 104691391.9204
R2: -0.7761
True Positions
Predicted Positions
X Position
Time Step
Y Position Comparison
RMSE: 101973551.9613
R2: -1.0275
Y Position
Z Position Comparison
RMSE: 6284163.1833
R2: -0.8748
Z Position

8. **Numerical Stability and Computational Considerations**
    - Throughout the process, we ensured numerical stability in calculations involving large or highly precise values.
    - We utilized NumPy's float128 data type for high precision, particularly during feature transformations and metric computations.
    - For convergence-requiring tasks like calculating anomalies, we employed iterative methods such as Newton-Raphson.

By systematically applying these methods, we have integrated classical numerical analysis techniques with modern machine learning approaches. This methodology demonstrates the robustness of computational tools in modelling and analysing planetary motion, highlighting the strengths and limitations of each approach.

3. Interpolation:

We firstly wanted to analyze which kind of interpolant would represent the true data with higher accuracy. We visualised both cubic and quadratic splines interpolation hypothesising what would suit better for orbital prediction. Since the revolution period of Venus is 225 days, we have trimmed down the input data for the interpolants to fall between [2021 Sep 17, 2022 Feb 27). Data points in this aphelion to aphelion time frame (1 Venus revolution) would be sufficient for interpolation, reducing the redundancy and model complexity. Cartesian data in the next following revolution time frame is considered as the true ground data.

Once we have got cubic and quadratic interpolation models for each axis namely X,Y and Z, derived by fitting over time array of length 255 (0 to 254), we went on to predict the position of Venus in between the sampled observation, to verify the continuity in the prediction.

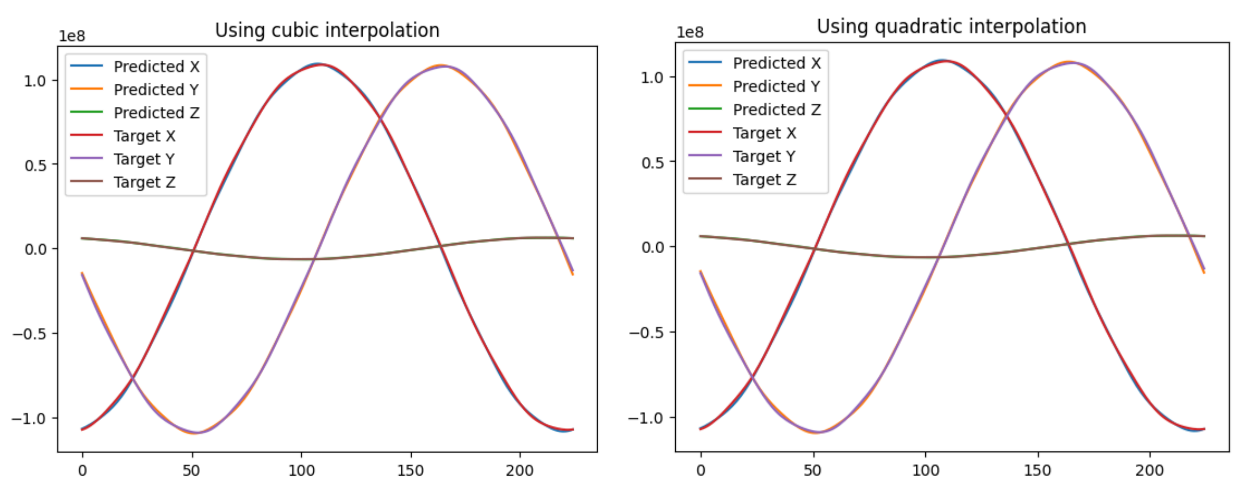
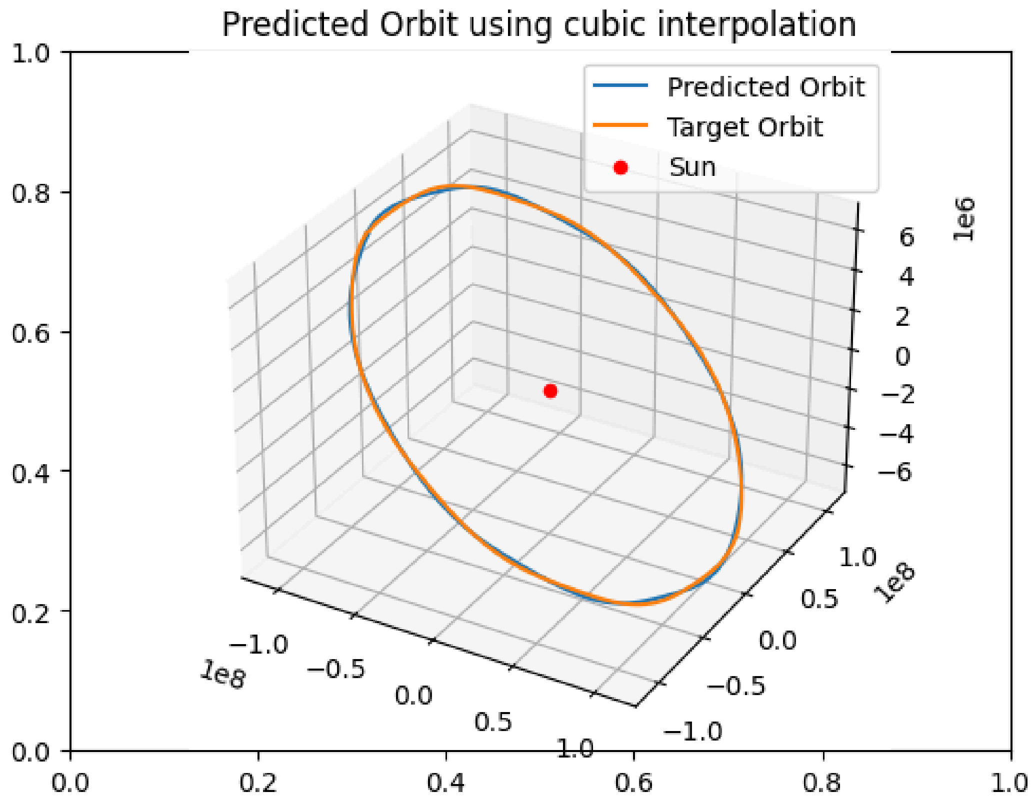


Figure 6. Plots of different kinds of interpolations, predicting the intermediate positions of Venus overlapped over observed true positions.


Predicted Orbit using cubic interpolation
Predicted Orbit
Target Orbit
Sun
1e6
1e8
1e8

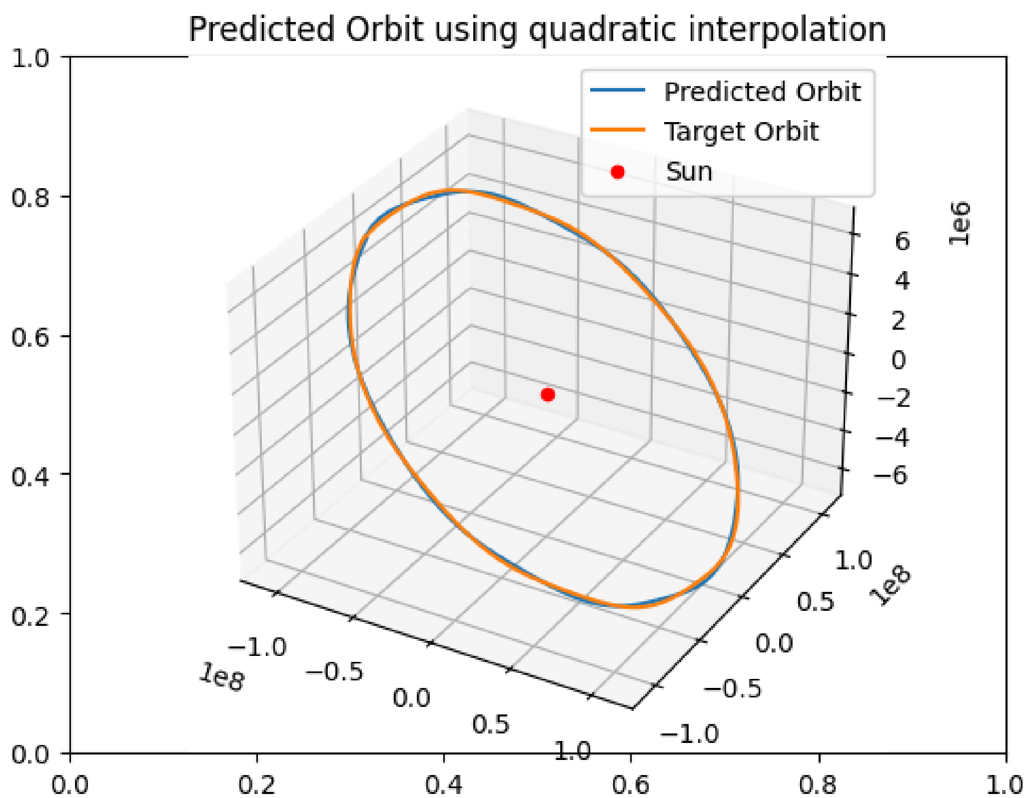
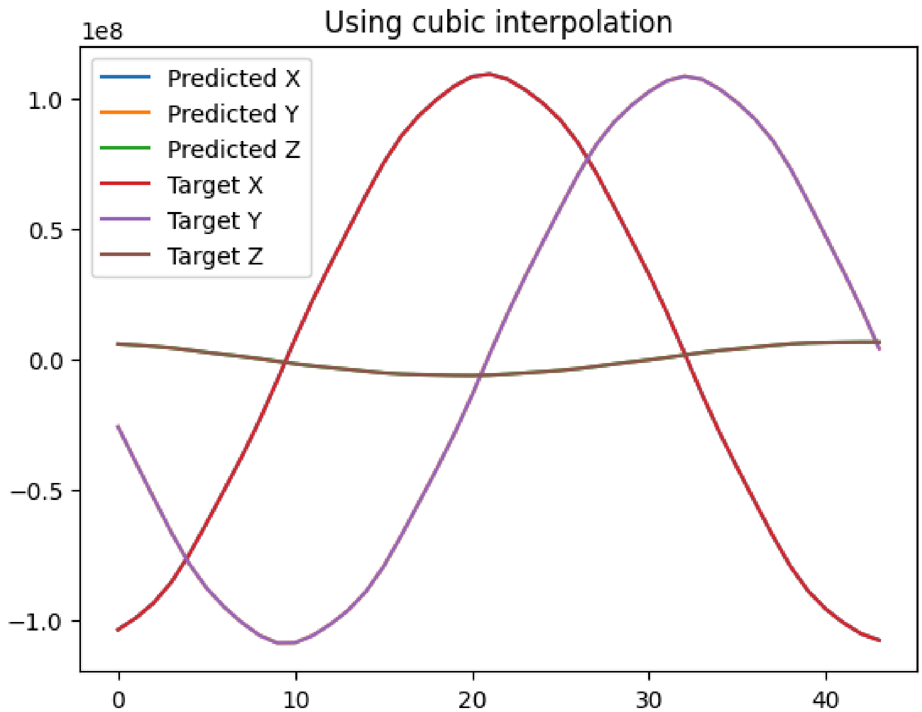

Predicted Orbit using quadratic interpolation
Predicted Orbit
Target Orbit
Sun
1e6
1e8
1e8

Figure 7. Visualisation of continuity in the orbits predicted by interpolation techniques.

To cross validate cubic splines interpolation, we have down-sampled the training input data to *80%* keeping the rest *20%* as test dataset. After modelling cubic splines over this volume of input data with 181 data points, we have predicted the other *20%* of the data (*44* data points). The predicted orbit almost overlaps the true data.



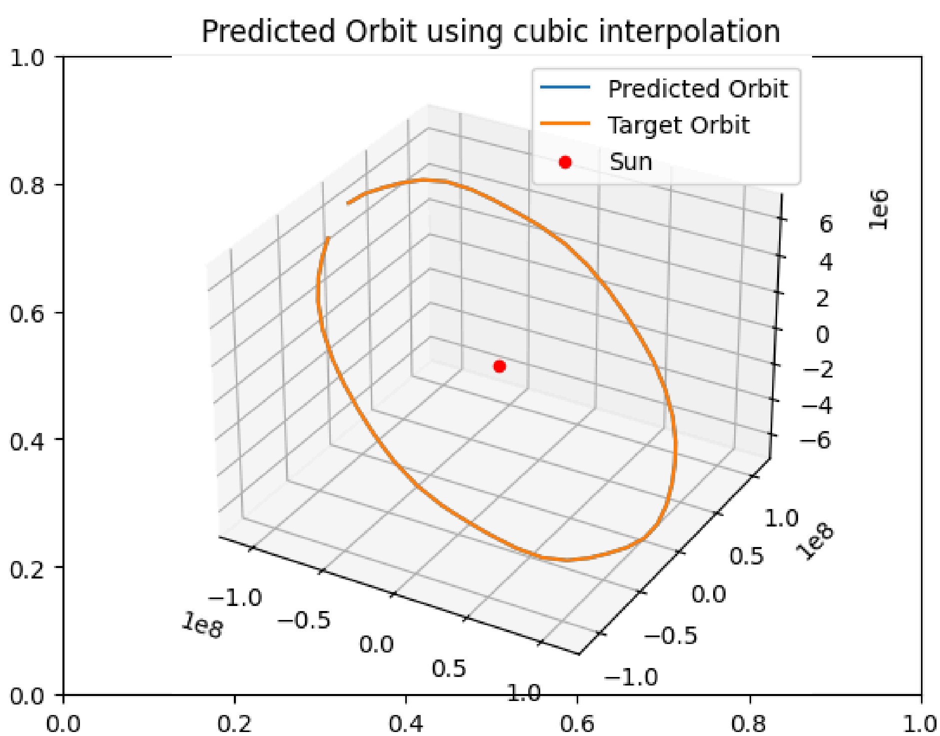


Figure 8. Plots showing robustness of cubic interpolation model.

Though the predictions seem to be very accurate in the visuals above, there is still a bit of error generated by the numerical instability when handling large numbers with high precision.

```
Unset
Mean Squared Error: 33398.95717705999 kms^2
Root Mean Squared Error: 182.75381576607364 kms
Mean Absolute Error: 246.89252215963708 kms
Final Error Percentage: 0.00017346684824155457%
```

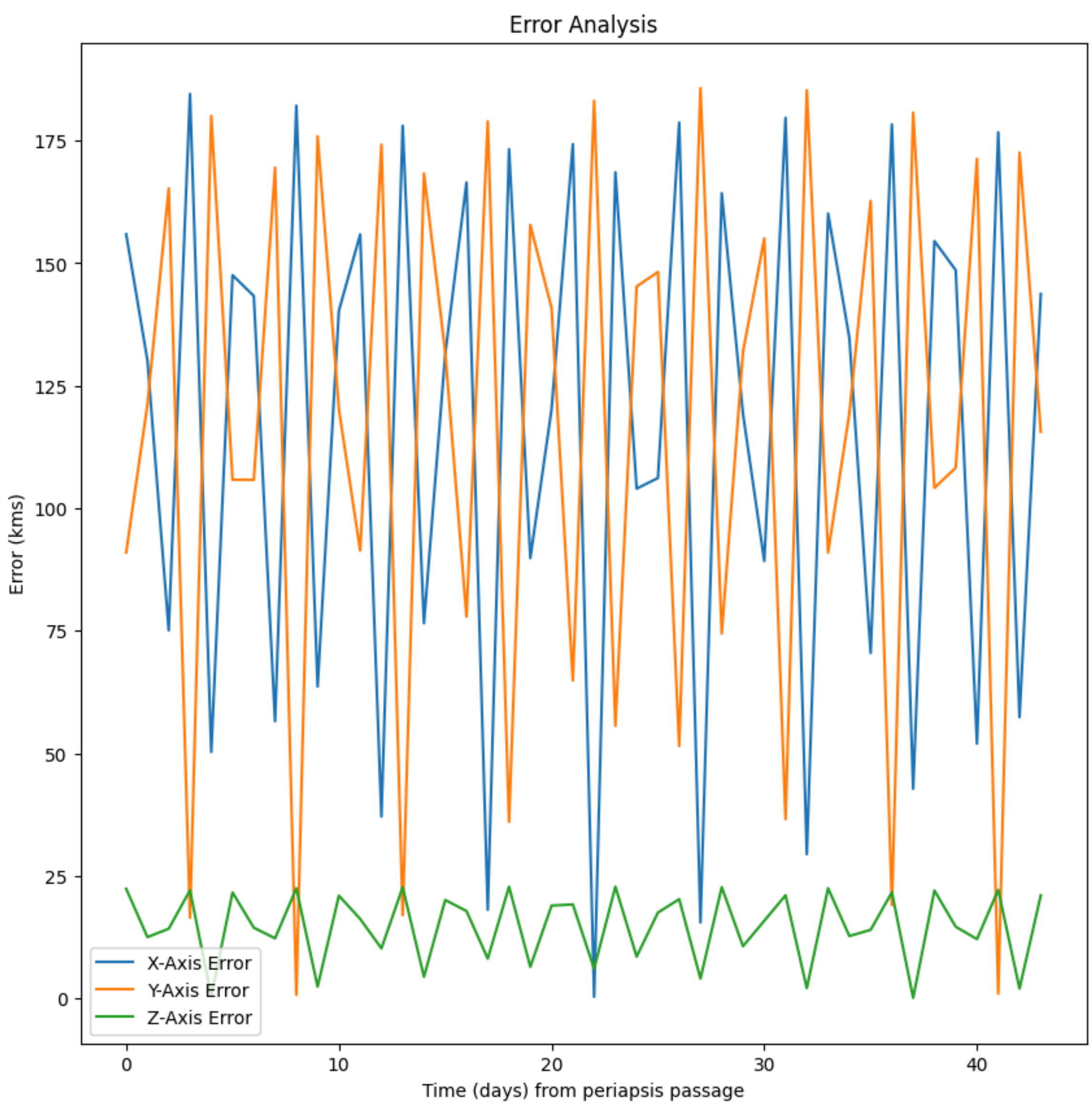


Figure 9. Error in interpolation technique is relatively lesser than the errors generated by other methods. The Error scale in Y-axis here is in kms, with the max value being in the neighbourhood of *170* kms.

5. Conclusion

This study explored three distinct methodologies—Kepler's equations, regression techniques, and interpolation approaches—for predicting Venus's orbit. While each method demonstrated unique strengths and limitations, their combined application underscores the versatility of numerical and data-driven techniques in celestial mechanics.

Kepler's equations, rooted in physics and celestial mechanics, provided a deterministic framework for deriving orbital parameters. However, challenges such as error propagation in eccentricity calculations and numerical instability highlighted the need for precision when handling astronomical data. This method demonstrated its strength in offering a deep understanding of the mechanics governing Venus's motion, making it invaluable for contexts requiring high fidelity and interpretability.

Regression techniques, enriched with trigonometric features to capture the periodic nature of planetary motion, leveraged modern machine learning tools to model orbital behavior. These methods excelled in handling large datasets and provided robust predictions across different coordinate axes. By employing advanced regression models and comprehensive error analysis, we demonstrated the adaptability of data-driven approaches to complement traditional physics-based techniques.

Interpolation methods bridged the gap between computational simplicity and predictive accuracy. Using cubic and quadratic splines, we generated continuous orbital paths, showcasing their utility in short-term predictions. They offer significant computational efficiency, making them practical for scenarios requiring rapid predictions with minimal computational overhead.

This project highlights the complementary nature of the above mentioned methodologies. Together, they form a toolkit that can be tailored to specific requirements, whether for academic research, space mission planning, or educational purposes.

This holistic approach demonstrates the importance of integrating domain knowledge with numerical and data-driven techniques. By doing so, we establish a framework for addressing similar challenges in planetary science and other fields reliant on numerical modeling.

References:

Horizons System https://ssd.jpl.nasa.gov/horizons/app.html#/,
https://ssd.jpl.nasa.gov/horizons/tutorial.html,
https://ssd.jpl.nasa.gov/horizons/manual.html#center

Python Scipy module
https://docs.scipy.org/doc/scipy/reference/generated/scipy.interpolate.interp1d.html

Cool cosmos
https://coolcosmos.ipac.caltech.edu/ask/54-how-long-does-it-take-venus-to-go-around-the-sun-#:~:text=Venus%20revolves%20or%20orbits%20around,once%20every%20224.7%20Earth%20days.

In-The-Sky https://in-the-sky.org/news.php?id=20210612_11_100