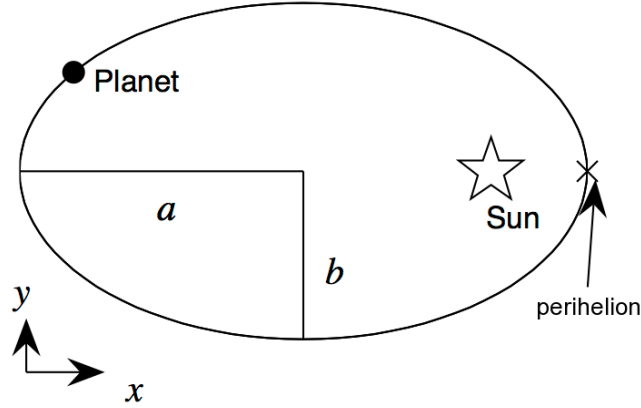


Computational Astrophysics

2019

Exercises 06. Root Finding

1. Eccentric Anomaly



A planet is moving around the sun in an elliptic Keplerian orbit with semi-major axis a , semi-minor axis b , and eccentricity $e = \sqrt{1 - b^2/a^2}$. The planet last perihelion occur at $t = 0$. The angular frequency of the motion is $\omega = 2\pi/T$, where T is the duration of its orbit (period).

If we define a 2D coordinate system (x, y) with origin at the center of the ellipse, then the points on the ellipse are described by the equation

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1 . \quad (1)$$

The location of the planet in the (x, y) coordinate system is given by

$$\begin{aligned} x &= a \cos E , \\ y &= b \sin E , \end{aligned} \quad (2)$$

where E is the *eccentric anomaly*, which is defined by

$$E = \omega t + e \sin E . \quad (3)$$

The interesting equation here is Eq. (3), which gives an *implicit* non-linear relationship for E . In order to find E for a given ωt and e , we will need to find the solution to the following equation:

$$0 = E - \omega t - e \sin E , \quad (4)$$

in other words, we have to find this equation's root!

- The Earth has an orbital period of 365.25635 days, a semi-major axis $a = 1.496 \times 10^8$ km = 1 AU, and its orbit has an eccentricity of $e = 0.0167$. Compute E , x and y for $t = 91$ days, $t = 182$ days, and $t = 273$ days using your favorite root finding method. The fractional error in E at the end of your computation (from one iteration to the next) should be less than 10^{-10} . How many iterations does your method need, i.e, how quickly does it converge?
- Now suppose that something (very bad) happened, putting Earth on a pretty eccentric orbit, say $e = 0.99999$. How many iterations does your algorithm need now? How could you accelerate convergence?

2. Polynomials with Multiple Roots

Find all real roots of

$$f(x) = 3x^5 + 5x^4 - x^3 . \quad (5)$$

How many are there and how can you construct an algorithm that finds all of them? This method should be general and should work for real roots of any polynomial in one variable with at least one real root. You may ask the internet for help!

Implement your method and be ready to be tested in class to demonstrate its capabilities with a few examples.

Happy Coding :) !