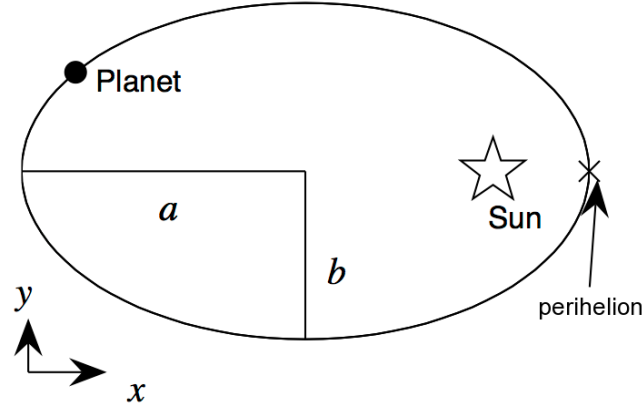


Computational Astrophysics

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 on 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.

Solution

1. (a) The function to be analyzed is

$$f(E) = E - e \sin E - \omega t. \quad (6)$$

In order to apply the NEWton-Raphson Method we need the derivative

$$\frac{df}{dE} = 1 - e \cos E. \quad (7)$$

Using these relations, the Newton-Raphson method for finding roots is implemented in solution7.1a.py. The resulting position coordinates and number of iterations for the times in the exercise are given in Table 1.

t (days)	E	x (AU)	y (AU)	Iterations
91	1.582092288991627	-0.011295721973131013	0.9997967554706361	3
182	3.1309642006817358	-0.9999435185260914	0.010626770643682799	3
273	4.679489100532153	-0.03289394502393073	-0.999319468509937	4
365.25635	6.283185307179586	1.0	-2.4489520327324266e - 16	4

Table 1: Results of the Newton-Raphson method finding the eccentric anomaly of the Earth

1. (b) In the code solution7.1b.py the eccentricity is changed to the value $e = 0.99999$ and the resulting values for the motion of the planet are presented in Table 2. Note that the number of iterations increased notably from point (a).

t (days)	E	x (AU)	y (AU)	Iterations
91	2.3066463874889322	-0.67121751444286	0.003315009203508128	13
182	3.1361896410659678	-0.9999854037633422	2.41628286026531e - 05	12
273	3.9636437776514932	-0.6807201024458955	-0.0032760264315865183	12
365.25635	6.283185307197494	1.0	8.008327018991285e - 14	23

Table 2: Results of the Newton-Raphson method finding the eccentric anomaly of the Earth

2. In order to obtain the roots of the polynomial, in solution7.2.py it is implemented the Weierstrass (or Druand-Kerner) algorithm. To define the procedure, consider the monic ¹ general polynomial of order 5

$$f(x) = x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0. \quad (8)$$

¹The coefficient of the highest degree term equals one.

Let the roots of this polynomial to be the complex numbers r_1, r_2, r_3, r_4, r_5 , so that the polynomial may be written as

$$f(x) = (x - r_1)(x - r_2)(x - r_3)(x - r_4)(x - r_5). \quad (9)$$

From this equation it is possible to isolate the root r_1 as

$$r_1 = x - \frac{f(x)}{(x - r_2)(x - r_3)(x - r_4)(x - r_5)}. \quad (10)$$

This equation may be used as an iteration relation to obtain the root if we begin with some approximated value $x^{(k)}$ and with approximated values for the other roots, $r_i^{(0)}$ for $(i = 2, 3, 4, 5)$, as

$$x^{(k+1)} = x^{(k)} - \frac{f(x)}{(x - r_2^{(0)})(x - r_3^{(0)})(x - r_4^{(0)})(x - r_5^{(0)})}. \quad (11)$$

The method proposed by Weierstrass (and later by Druand and Kerner) is to combine this iteration with similar relations for the other roots, i.e.

$$r_1^{(k)} = r_1^{(k-1)} - \frac{f(r_1^{(k-1)})}{(r_1^{(k-1)} - r_2^{(k-1)})(r_1^{(k-1)} - r_3^{(k-1)})(r_1^{(k-1)} - r_4^{(k-1)})(r_1^{(k-1)} - r_5^{(k-1)})} \quad (12)$$

$$r_2^{(k)} = r_2^{(k-1)} - \frac{f(r_2^{(k-1)})}{(r_2^{(k-1)} - r_1^{(k-1)})(r_2^{(k-1)} - r_3^{(k-1)})(r_2^{(k-1)} - r_4^{(k-1)})(r_2^{(k-1)} - r_5^{(k-1)})} \quad (13)$$

$$r_3^{(k)} = r_3^{(k-1)} - \frac{f(r_3^{(k-1)})}{(r_3^{(k-1)} - r_1^{(k-1)})(r_3^{(k-1)} - r_2^{(k-1)})(r_3^{(k-1)} - r_4^{(k-1)})(r_3^{(k-1)} - r_5^{(k-1)})} \quad (14)$$

$$r_4^{(k)} = r_4^{(k-1)} - \frac{f(r_4^{(k-1)})}{(r_4^{(k-1)} - r_1^{(k-1)})(r_4^{(k-1)} - r_2^{(k-1)})(r_4^{(k-1)} - r_3^{(k-1)})(r_4^{(k-1)} - r_5^{(k-1)})} \quad (15)$$

$$r_5^{(k)} = r_5^{(k-1)} - \frac{f(r_5^{(k-1)})}{(r_5^{(k-1)} - r_1^{(k-1)})(r_5^{(k-1)} - r_2^{(k-1)})(r_5^{(k-1)} - r_3^{(k-1)})(r_5^{(k-1)} - r_4^{(k-1)})}, \quad (16)$$

or in general,

$$r_i^{(k)} = r_i^{(k-1)} - \frac{f(r_i^{(k-1)})}{\prod_{j \neq i} (r_i^{(k-1)} - r_j^{(k-1)})} \quad (17)$$

for $i = 1, 2, \dots, n$.

[1] http://homepages.math.uic.edu/~jan/mcs471f03/Project_Two/proj2/node2.html

[2] https://en.wikipedia.org/wiki/Durand%E2%80%93Kerner_method