WS5: Root Finding

Jackie Villadsen

January 28, 2012

1 Eccentric Anomaly

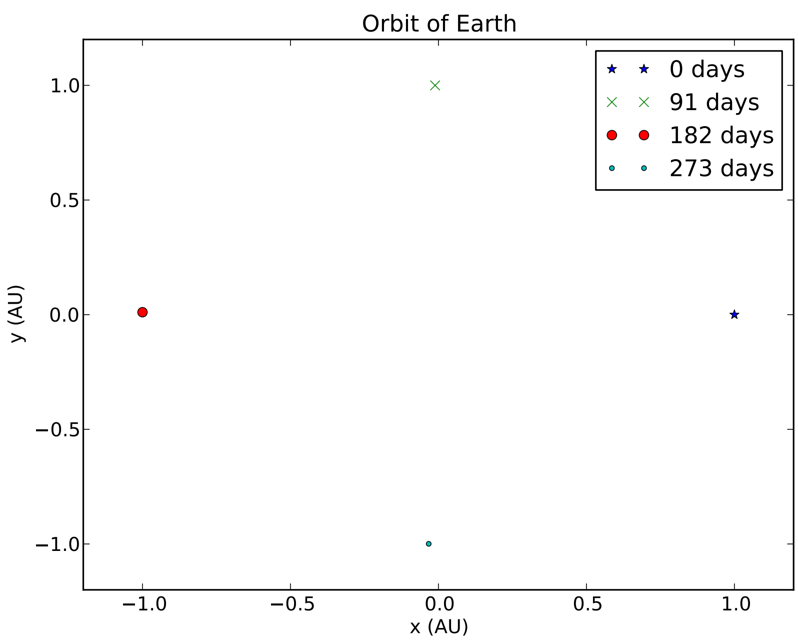

Figure 1: The position of Earth in its orbit at 0, 91, 182, and 273 days, for its current eccentricity of 0.0167.

I used Newton's method to solve for the eccentric anomaly at different times in Earth's orbit. Figure 1 shows the positions of Earth at t=0 days (perihelion), 91 days, 182 days, and 273 days, for its current eccentricity of 0.0167. Figure 2 shows the position of Earth at the same times if its eccentricity were changed to 0.99999. Table 1 shows the number of iterations needed to converge to a

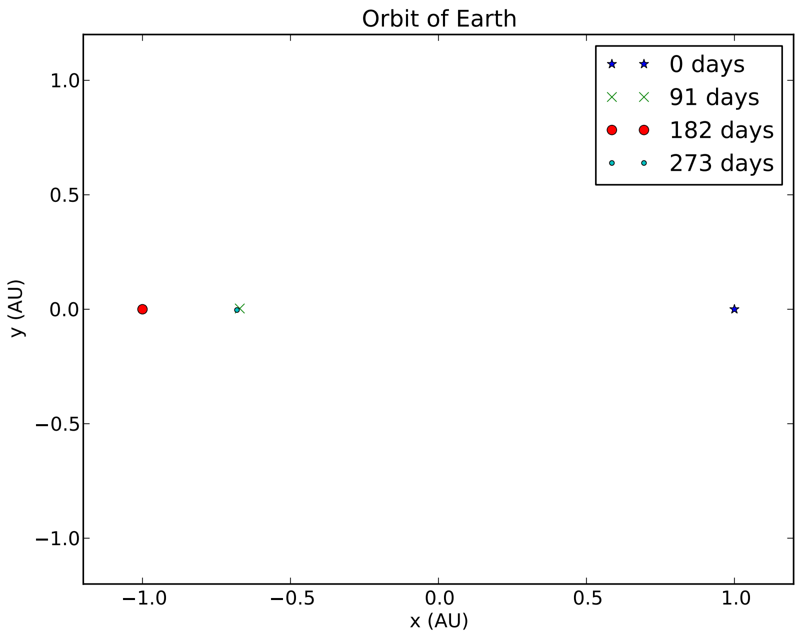

Figure 2: The position of Earth in its orbit at 0, 91, 182, and 273 days, if its eccentricity were changed to 0.99999.

fractional error less than 10^{-10} for each eccentricity, depending on the initial guess used for E. For $E_0 = 0$, the number of iterations increased a lot with the eccentricity. However, for $E_0 = \Omega t$, the number of iterations remained quite low even for high eccentricity.

2 Polynomials with Multiple Roots

I used the Durand-Kerner Method to solve for all the complex roots of any polynomial simultaneously. The Durand-Kerner method relies on the fact that the polynomial can be factored into the form:

$$f(x) = \prod_i (x - r_i) \tag{1}$$

where r_i are the roots of the polynomial.

To obtain the n roots of a polynomial of degree n, I started with guesses: $r_i|_0 = (0.4 + 0.9j)^i$, as suggested by Wikipedia. These initial guesses are arbitrary but fulfill the requirement that only one of them is a real number or a root of unity (i=0).

I then iterated forwards using the expression below, for 100 iterations (I found it difficult to set up a convergence criterion for complex numbers, so instead I picked an arbitrary large number of iterations):

$$r_i|_{n+1} = r_i|_n - \frac{f(r_i|_n)}{\prod_{i \neq j}(r_i|_n - r_j|_n)} \tag{2}$$

For the polynomial given in the worksheet, $f(x) = 3x^5 + 5x^4 - x^3$, I obtained three roots whose real and imaginary components were of order 10^{-18}, corresponding to the three zero roots of the polynomial. The other two roots were 0.1805 and -1.847 (with imaginary parts of order 10^{-49} or less), which are the roots of the quadratic obtained by dividing by x^3.

I also tested my algorithm on a 3rd-degree polynomial given on Wikipedia, $f(x) = x^3 - 3x^2 + 3x - 5$, successfully recovering the one real root, 2.5874, and the two imaginary roots, 0.206±1.375j. Finally, I tested my algorithm on the simple example $f(x) = x^2$, which yielded two roots whose complex and imaginary parts were both of order 10^{-30} (essentially zero).

Time t (days)	$e = 0.0167, E_0 = 0$	$e = 0.0167, E_0 = \Omega t$	$e = 0.99999, E_0 = 0$	$e = 0.99999, E_0 = \Omega t$
0	1	1	1	1
91	5	4	68	6
182	4	3	35	3
273	5	4	106	6

Table 1: This table shows the number of iterations needed to converge on the root E(t) using Newton's method, for low and high eccentricity, and for using an initial guess of E=0 vs. a more intelligent initial guess of $E = \Omega t$.