UNIFORM ALGORITHM FOR RECTILINEAR TWO-BODY MOTION

Roger L. Mansfield, 2002 February 15
http://astroger.com

(Updated to PTC's Mathcad Prime 10.0 on 2024 July 23)

In issue #43 of *The Orrery,* Greg Neill poses and answers the question, "If Earth were to stop in its orbit and fall into the Sun, how long would it fall?" [1]. This called to my mind a uniform, two-body theory that I presented at an AIAA conference in 1986 [2]. In this worksheet we consider "Earth falls into to the Sun" as an application of that theory. We take Earth and the Sun to be gravitating point masses, or "Newtonian particles", in accordance with the assumptions.

Before we construct our application, we need to define the c-functions $c_1(x)$, $c_2(x)$, and $c_3(x)$ in a way that will be convenient for use in this worksheet.

$$c_1(x) := \begin{array}{l} \text{if } |x| < 10^{-10} \\ \quad c_1 \leftarrow 1 \\ \text{else} \\ \quad \text{if } x > 0 \\ \quad\quad c_1 \leftarrow \dfrac{\sin\left(\sqrt{x}\right)}{\sqrt{x}} \\ \quad \text{else} \\ \quad\quad c_1 \leftarrow \dfrac{\sinh\left(\sqrt{-x}\right)}{\sqrt{-x}} \\ c_1 \end{array}$$

$$c_2(x) := \begin{array}{l} \text{if } |x| < 10^{-10} \\ \quad c_2 \leftarrow 0.5 \\ \text{else} \\ \quad \text{if } x > 0 \\ \quad\quad c_2 \leftarrow \dfrac{1-\cos\left(\sqrt{x}\right)}{x} \\ \quad \text{else} \\ \quad\quad c_2 \leftarrow \dfrac{1-\cosh\left(\sqrt{-x}\right)}{x} \\ c_2 \end{array}$$

$$c_3(x) := \begin{array}{l} \text{if } |x| < 10^{-10} \\ \quad c_3 \leftarrow \dfrac{1}{6} \\ \text{else} \\ \quad \text{if } x > 0 \\ \quad\quad c_3 \leftarrow \dfrac{\sqrt{x}-\sin\left(\sqrt{x}\right)}{x \cdot \sqrt{x}} \\ \quad \text{else} \\ \quad\quad c_3 \leftarrow \dfrac{\sqrt{-x}-\sinh\left(\sqrt{-x}\right)}{x \cdot \sqrt{-x}} \\ c_3 \end{array}$$

These functions take advantage of the fact that Stumpff's c-functions have familiar circular and hyperbolic function representations. But note that, in general, one should use an algorithm such as the Mathcad procedural function **C**(x), as given in the Appendix. (Function **C**(x) employs series and recursion, and does not need to test whether the argument x is positive, zero, or negative.)

We set up the constants for the Sun-Earth two-body system, for the case of "Earth falls into the Sun" from x = -1 A.U., and with zero initial orbital velocity in the heliocentric ecliptic plane.

$k := 0.01720209895$ Gaussian constant in A.U.$^{3/2}$ / day.

$\mu := 1.00000304$ Mass of Sun + Earth, in solar masses.

$K := k \cdot \sqrt{\mu}$ Two-body constant for Sun-Earth system.

$r_o := 1.0$ Earth is assumed to be at 1 A.U. at time of fall.

$h := 0$ h is the magnitude of the specific angular momentum vector, **h**, defined as **h** = **r** x **v**, where **r** is the position vector, with components in A.U., and **v** is the velocity vector, with components in A.U./day.

$\alpha := \dfrac{2 \cdot K^2}{r_o}$ The energy parameter, α, is a function of the initial radius vector magnitude only, since the initial velocity is assumed to be zero. But in general, since α is -2 times the total specific mechanical energy of the two-body system, we have $\alpha = 2K^2/r - \mathbf{v}\cdot\mathbf{v}$.

$ecc := \sqrt{1 - \dfrac{\alpha \cdot h^2}{K^4}}$ Quantity "ecc" is the orbital eccentricity. It is 1 for all three possible rectilinear paths: rectilinear ellipse (α > 0, h = 0), rectilinear parabola (α = 0, h = 0), and rectilinear hyperbola (α < 0, h = 0).

$p := \dfrac{h^2}{K^2}$ p is the semilatus rectum. It is zero for a rectilinear path.

$q := \dfrac{p}{1 + ecc}$ q is the periapsidal distance. It is zero for a rectilinear path.

We can now give the equations for the perifocal* x and y components of Earth's position in the orbital plane, for the radius vector magnitude, and for time as a function of the "fictitious time", s. We should note that s is defined by the Sundmann transformation dt/ds = r, i.e., t is the integral of r ds.

$$x(s) := q - K^2 \cdot s^2 \cdot c_2(\alpha \cdot s^2) \qquad y(s) := K \cdot \sqrt{q \cdot (1 + ecc)} \cdot s \cdot c_1(\alpha \cdot s^2)$$

$$r(s) := q + K^2 \cdot ecc \cdot s^2 \cdot c_2(\alpha \cdot s^2) \qquad t(s) := q \cdot s + K^2 \cdot ecc \cdot s^3 \cdot c_3(\alpha \cdot s^2)$$

*"Perifocal" means that the x-axis lies along the line of apsides and the direction of positive x is from the origin toward periapsis. In the rectilinear case, periapsis is at the origin and all values of perifocal x must be zero or negative; all values of perifocal y must be zero.

It is now a simple matter to calculate the time it would take for Earth to fall into the Sun. Since for elliptical motion the eccentric anomaly E = $\alpha^{1/2}$ s, and since we want to fall for half an orbit (a full orbit would have Earth bouncing off the Sun and coming back out to r = 1 A.U.), we set E equal to the number of radians in 180 degrees, calculate s from E, and calculate t from s:

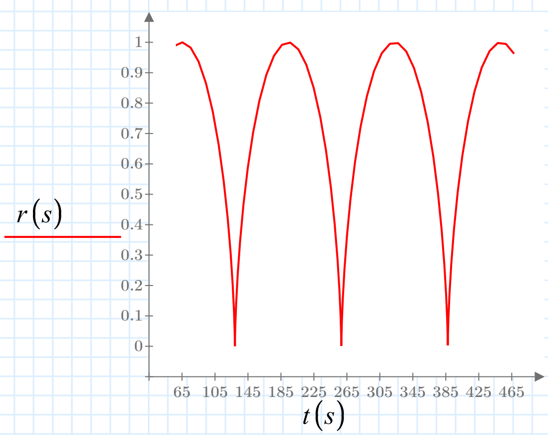
$$DegPerRad := \frac{180}{\pi} \qquad E := \frac{180}{DegPerRad}$$

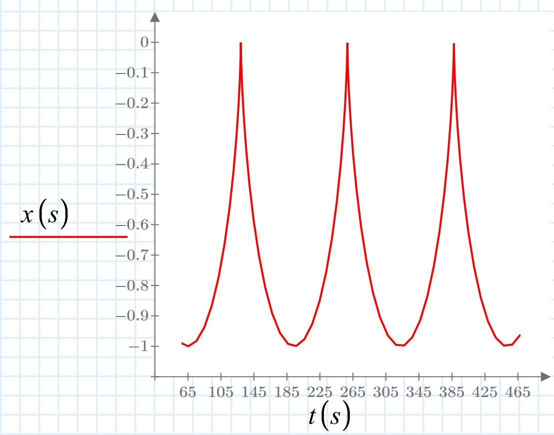
$$s_1 := \frac{E}{\sqrt{\alpha}} \qquad t_1 := t(s_1)$$

$t_1 = 64.56880928$ (days)

We plot r and x as functions of t(s) for three "bounces" of Earth off the Sun. The units of r are A.U. and the units of t are days. First we set up a range of s values as needed for the plots. We start the plots at t_1 because the equations for r, x, and y assume that s and t are zero at periapsis.

$s := 0, 10 .. 1000$





In the appendix below we define and test Mathcad procedural function, **C**(x). **C**(x) is based upon an algorithm constructed by Danby [3] to compute the first four c-functions of argument x by series and recursion.

REFERENCES

[1] Neill, Greg, "Taking the Fall," The Orrery, Issue #43 (December 2001), pp. 13-15.

[2] Mansfield, Roger L., "Uniform, Non-Singular Path Representation for Highly Energetic Space Objects," *AIAA/AAS Paper No. 86-2269-CP,* Williamsburg, Virginia (August 1986).

[3] Danby, J.M.A., *Fundamentals of Celestial Mechanics,* Second Edition (1988), Willmann-Bell, Richmond Virginia, p. 173 (http://www.willbell.com).

APPENDIX

The following Mathcad procedural function computes the first four c-functions c_0, c_1, c_2, and c_3 of argument x. Since we have already defined representations for c_1, c_2, and c_3 above, it is a simple matter now to check the c-functions computed by series and recursion against the c-functions computed via their circular and hyperbolic representations.

$$
\begin{aligned}
C(x) := \; & N \leftarrow 0 \\
& \text{while } |x| \geq 0.1 \\
& \quad x \leftarrow \frac{x}{4} \\
& \quad N \leftarrow N+1 \\
& c_3 \leftarrow \frac{\left(1-\frac{x}{20}\cdot\left(1-\frac{x}{42}\cdot\left(1-\frac{x}{72}\cdot\left(1-\frac{x}{110}\cdot\left(1-\frac{x}{156}\cdot\left(1-\frac{x}{210}\right)\right)\right)\right)\right)\right)}{6} \\
& c_2 \leftarrow \frac{\left(1-\frac{x}{12}\cdot\left(1-\frac{x}{30}\cdot\left(1-\frac{x}{56}\cdot\left(1-\frac{x}{090}\cdot\left(1-\frac{x}{132}\cdot\left(1-\frac{x}{182}\right)\right)\right)\right)\right)\right)}{2} \\
& c_1 \leftarrow 1 - c_3 \cdot x \\
& c_0 \leftarrow 1 - c_2 \cdot x \\
& \text{while } N > 0 \\
& \quad N \leftarrow N-1 \\
& \quad c_3 \leftarrow \frac{\left(c_1 \cdot c_2 + c_3\right)}{4} \\
& \quad c_2 \leftarrow \frac{c_1 \cdot c_1}{2} \\
& \quad c_1 \leftarrow c_1 \cdot c_0 \\
& \quad c_0 \leftarrow 2 \cdot c_0 \cdot c_0 - 1 \\
& c
\end{aligned}
$$

Here is our check. (We don't need c_0, but it is computed by **C** nevertheless.)

c_0 - c_3 by series and recursion:

$$C(0.5)=\begin{bmatrix}0.7602446\\0.91872537\\0.47951081\\0.16254926\end{bmatrix} \qquad C(-0.5)=\begin{bmatrix}1.26059184\\1.08544164\\0.52118367\\0.17088328\end{bmatrix}$$

c_1 - c_3 via circular and hyperbolic functions:

$$c_1(0.5)=0.91872537 \qquad c_1(-0.5)=1.08544164$$

$$c_2(0.5)=0.47951081 \qquad c_2(-0.5)=0.52118367$$

$$c_3(0.5)=0.16254926 \qquad c_3(-0.5)=0.17088328$$