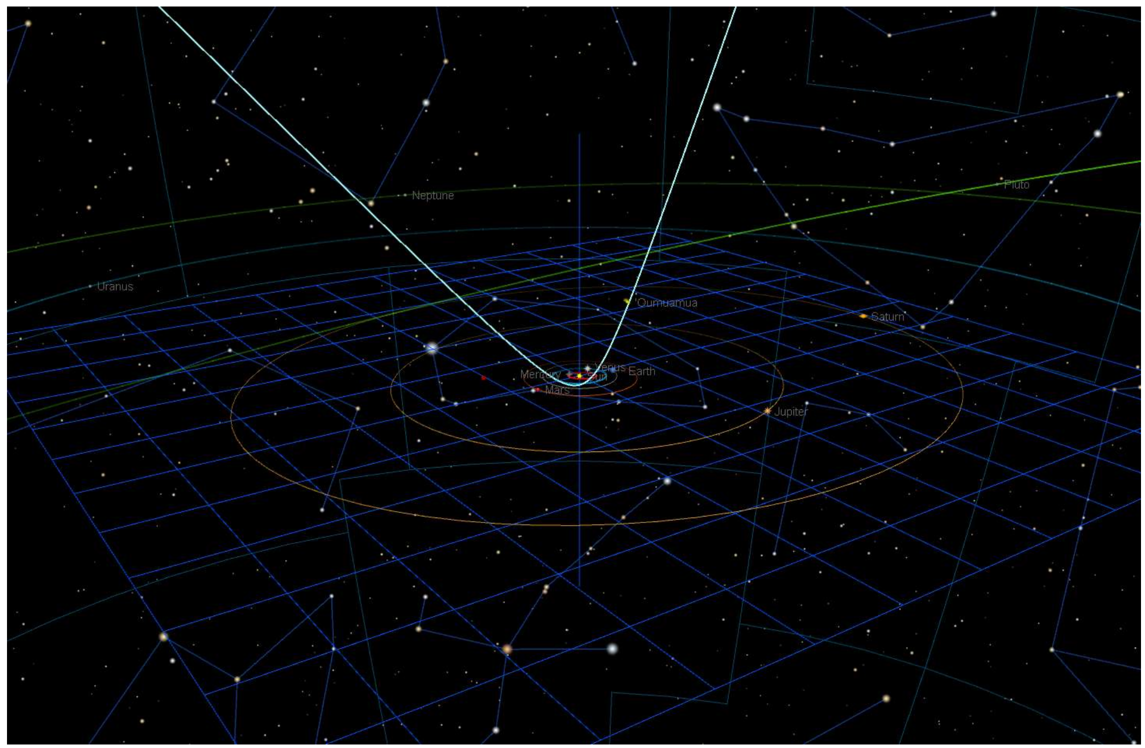
WHAT ARE UNIVERSAL VARIABLES? ***How Do They Describe the Path of the First-Known Interstellar Asteroid, `Oumuamua?***

by Roger L. Mansfield, *MAA 50-Year Member*
Astronomical Data Service, Colorado Springs, Colorado USA
http://astroger.com

Presentation for the 2019 meeting of the Rocky Mountain Section (RMS) *of the* Mathematical Association of America (MAA)

Fort Lewis College
Durango, Colorado

April 5-6, 2019



The figure above depicts the position of` `Oumuamua on its hyperbolic path, three months before its perihelion on 2017 September 9

Universal Variables Theories

- The eccentricity e of a two-body orbit or path indicates whether the motion is along an ellipse ($e < 1$), a parabola ($e = 1$), or a hyperbola ($e > 1$)
- In a "universal variables" theory, there is one set of equations that describes all of the possible conic paths of two-body motion
- Richard H. Battin [3], William H. Goodyear [4], and Samuel Herrick [5] all published universal variables theories of two-body motion in the 1960s
- But Karl J. Stumpff [1] had already published such a theory in 1947, using a new class of transcendental functions that have come to be called the Stumpff functions [6] or "c-functions"
- During this presentation, we will examine the vector equations for the propagation of a two-body path along an ellipse, a parabola, or a hyperbola, using a universal, two-body theory that traces back to Stumpff's seminal article in the *Astronomische Nachrichten* in 1947

Outline and Objective of this Presentation

- **We view a depiction of the hyperbolic trajectory of the first-known interstellar asteroid, `Oumuamua, as an application of the vector equations of motion that we had just discussed**

- **We will then examine the first six c-functions in some detail**
  - **Series definitions and recurrence formulae**
  - **Derivatives and integrals**
  - **Quadruple-argument formulae**
  - **Calculation by series and recursion**

- **My objective for this presentation is that you will:**
  - **understand and appreciate the importance of Stumpff's c-functions in applied mathematics**
  - **understand how to calculate the first six c-functions**

- **This presentation is available at my website at http://astroger.com, along with a Borland C++ Builder 5 program that propagates the trajectory of `Oumuamua (and thus includes code that calculates the first six c-functions)**

Excerpt from Stumpff's "Neue Formeln" Article [1]

und ist stets reell, da ja E und M stets gleichzeitig reell oder imaginär sind. Die Funktionen c_2 und c_3 entstammen der Folge

$$\left.\begin{aligned}
c_0 &= \cos\lambda \\
c_1 &= \frac{1}{\lambda}\int_0^\lambda \cos\lambda\, d\lambda = \frac{\sin\lambda}{\lambda} \\
c_2 &= \frac{1}{\lambda^2}\int_0^\lambda\int_0^\lambda \cos\lambda\,(d\lambda)^2 = \frac{1-\cos\lambda}{\lambda^2} \\
&\dots \\
c_n &= \frac{1}{\lambda^n}\int_0^\lambda\dots\int_0^\lambda \cos\lambda\,(d\lambda)^n = \frac{1}{n!}\left[1-\frac{\lambda^2}{(n+1)(n+2)}+\right. \\
&\qquad\left.+\frac{\lambda^4}{(n+1)\cdots(n+4)}\dots\right]
\end{aligned}\right\} \quad (10)$$

und sind als Funktionen von λ^2 stets reell, auch wenn man für λ^2 negative Werte (Hyperbeln) oder null (Parabeln) zuläßt.

Second-Order ODE for Two-Body (Unperturbed) Orbital Motion

$$\frac{d^2\mathbf{r}}{dt^2} + K^2 \frac{\mathbf{r}}{r^3} = 0$$ (Here 3-vectors are boldface type)

For perturbed orbital motion, $\frac{d^2\mathbf{r}}{dt^2} + K^2 \frac{\mathbf{r}}{r^3} = P\,(r\,, \frac{d\mathbf{r}}{dt})$, where $\mathbf{P}$ is a 3-vector of perturbative accelerations. In both equations now,

$\mathbf{r}$ (boldface) is position vector of secondary, with components x, y, and z

$\frac{d\mathbf{r}}{dt}$ is velocity vector of secondary, with components $\frac{dx}{dt}$, $\frac{dy}{dt}$, and $\frac{dz}{dt}$

r (not boldface) is the magnitude of vector $\mathbf{r}$. That is, $r = \sqrt{\,}(x^2 + y^2 + z^2)$

K is gravity constant , i.e., $K = k \sqrt{\,}(1 + m/M)$, where k is Gaussian constant for primary, m is mass of secondary and M is mass of primary

Note: Next slide uses overscript dot ($\dot{}$) instead of $d\,/\,d\,t$ to denote derivative!

Two-Body Orbit Propagation using Stumpff's c-Functions

Given $\mathbf{r}_o$ and $\dot{\mathbf{r}}_o$ at time t_o, find $\mathbf{r}$ and $\dot{\mathbf{r}}$ at t as follows.

Compute $\alpha = 2/r_o - (\dot{\mathbf{r}}_o \cdot \dot{\mathbf{r}}_o)$; $\sigma_o = \mathbf{r}_o \cdot \dot{\mathbf{r}}_o$; $\tau = K(t - t_o)$.

Then solve Stumpff's equation [1],

$$\tau = r_o s c_1(\alpha s^2) + \sigma_o s^2 c_2(\alpha s^2) + s^3 c_3(\alpha s^2),$$

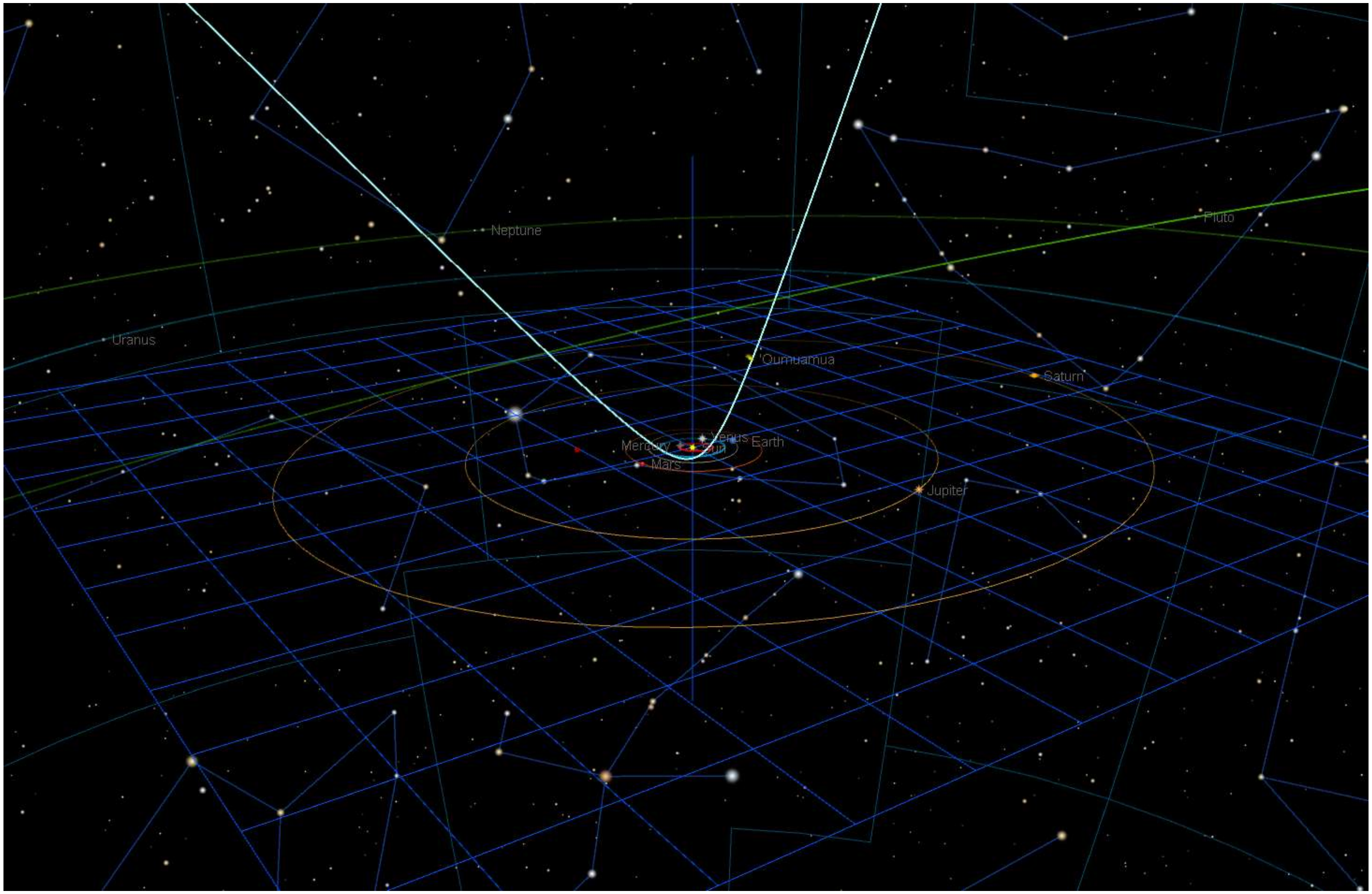
for s and r, using the fact that $d\tau/ds = r$. Finally,

$$f = 1 - s^2 c_2(\alpha s^2)/r_o \qquad g = \tau - s^3 c_3(\alpha s^2)$$

$$\dot{f} = -s c_1(\alpha s^2)/(r r_o) \qquad \dot{g} = 1 - s^2 c_2(\alpha s^2)/r$$

$$\mathbf{r} = f\,\mathbf{r}_o + g\,\dot{\mathbf{r}}_o \qquad \dot{\mathbf{r}} = \dot{f}\,\mathbf{r}_o + \dot{g}\,\dot{\mathbf{r}}_o$$

The Hyperbolic Trajectory of `Oumuamua



The figure above depicts the position of` `Oumuamua on its hyperbolic path, three months before its perihelion on 2017 September 9. It was created using Software Bisque's TheSky

Newton-Raphson Solution of Stumpff's Equation

First guess: $s^{(k)} = \tau/r_o$, for $k = 0$

Iterate on: $s^{(k+1)} = s^{(k)} - f(s^{(k)})/[df(s^{(k)})/ds]$

where $f(s) = r_o s c_1(\alpha s^2) + \sigma_o s^2 c_2(\alpha s^2) + s^3 c_3(\alpha s^2) - \tau$

and $df(s)/ds = r_o c_0(\alpha s^2) + \sigma_o s c_1(\alpha s^2) + s^2 c_2(\alpha s^2)$

$= r$

For Laguerre–Conway second–order solution [2], need second derivative, $f''(s)$:

$$f''(s) = dr/ds = -r_o \alpha s c_1(\alpha s^2) + \sigma_o c_0(\alpha s^2) + s c_1(\alpha s^2)$$

Series Definitions of c-Functions

$$c_n(\lambda^2) = \Sigma_{k=0}^{\infty}(-1)^k\lambda^{2k}/(2k+n)! \quad (n = 0,\ 1,\ 2,\ \ldots\ )$$

$$c_0(\lambda^2) = \frac{1}{0!} - \frac{\lambda^2}{2!} + \frac{\lambda^4}{4!} - \frac{\lambda^6}{6!} + \ldots$$

$$c_1(\lambda^2) = \frac{1}{1!} - \frac{\lambda^2}{3!} + \frac{\lambda^4}{5!} - \frac{\lambda^6}{7!} + \ldots$$

$$c_2(\lambda^2) = \frac{1}{2!} - \frac{\lambda^2}{4!} + \frac{\lambda^4}{6!} - \frac{\lambda^6}{8!} + \ldots$$

$$c_3(\lambda^2) = \frac{1}{3!} - \frac{\lambda^2}{5!} + \frac{\lambda^4}{7!} - \frac{\lambda^6}{9!} + \ldots$$

$$c_4(\lambda^2) = \frac{1}{4!} - \frac{\lambda^2}{6!} + \frac{\lambda^4}{8!} - \frac{\lambda^6}{10!} + \ldots$$

$$c_5(\lambda^2) = \frac{1}{5!} - \frac{\lambda^2}{7!} + \frac{\lambda^4}{9!} - \frac{\lambda^6}{11!} + \ldots$$

Recurrence Formulae for c-Functions

$$c_n(\lambda^2) = \frac{1}{n!} - \lambda^2 c_{n+2}(\lambda^2)$$

for $n = 0, 1, 2, \ldots$

Follows by inspection, or formally, by "index manipulation" on the defining series.

Therefore, given c_5 and c_4 computed by series, it is possible to "proceed downward in n" to calculate c_3, c_2, c_1, and c_0.

Can also "proceed upward in n" from c_0 and c_1. This permits greater efficiency of calculation in some cases.

Integral Definitions of c-Functions

$$c_n(\lambda^2) = \frac{1}{\lambda^n} \int_0^\lambda \lambda^{n-1} c_{n-1}(\lambda^2)\, d\lambda \qquad (n = 0, 1, 2, \ldots)$$

<u>For $\lambda^2 \geq 0$</u>

$c_0(\lambda^2) = \cos \lambda$

$c_1(\lambda^2) = (\sin \lambda)/\lambda$

$c_2(\lambda^2) = (1 - \cos \lambda)/\lambda^2$

$c_3(\lambda^2) = (\lambda - \sin \lambda)/\lambda^3$

<u>For $-\lambda^2 \leq 0$</u>

$c_0(-\lambda^2) = \cosh \lambda$

$c_1(-\lambda^2) = (\sinh \lambda)/\lambda$

$c_2(-\lambda^2) = (\cosh \lambda - 1)/\lambda^2$

$c_3(-\lambda^2) = (\sinh \lambda - \lambda)/\lambda^3$

and for $\lambda^2 = 0$, $c_n(\lambda^2) = 1/n!$

Differentiation of c-Functions

(1) For $n > 0$, $$\frac{d\lambda^n c_n(\lambda^2)}{d\lambda} = \lambda^{n-1} c_{n-1}(\lambda^2).$$

(2) For $n = 0$, $$\frac{dc_0(\lambda^2)}{d\lambda} = -\lambda c_1(\lambda^2) \qquad (*).$$

Equation (2) follows from Eq. (1) and the recurrence formulae.

*But note that $\frac{dc_0(\alpha s^2)}{ds} = -\alpha s c_1(\alpha s^2)$.

Quadruple-Argument Formulae for c-Functions

The following formulae are needed in order to construct an efficient algorithm for calculation of c_0, c_1, c_2 c_3, c_4, and c_5.

$$c_0(4\lambda^2) = 2c_0(\lambda^2)c_0(\lambda^2) - 1$$

$$c_1(4\lambda^2) = c_0(\lambda^2)c_1(\lambda^2)$$

$$c_2(4\lambda^2) = [c_1(\lambda^2)c_1(\lambda^2)]/2$$

$$c_3(4\lambda^2) = [c_1(\lambda^2)c_2(\lambda^2) + c_3(\lambda^2)]/4$$

$$c_4(4\lambda^2) = [c_2(\lambda^2)c_2(\lambda^2) + 2c_4(\lambda^2)]/8$$

$$c_5(4\lambda^2) = [c_2(\lambda^2)c_3(\lambda^2) + c_4(\lambda^2) + c_5(\lambda^2)]/16$$

Calculation of c-Functions

Need c_0, c_1, c_2 c_3, c_4, and c_5 to compute state transition matrix. Need an efficient procedure for calculating all six of these c–functions, valid for any real argument:

- Reduce argument of c–functions N times ($N \geq 0$) by quartering, until reduced argument is acceptably small*.
- Calculate c_5 and c_4 of reduced argument using series, carrying as many terms as needed for desired precision*.
- Calculate c_3, c_2, c_1, and c_0 of reduced argument via recurrence formulae.

*For this application, $|\lambda^2| < 0.1$ was "smallness criterion", and seven terms were used in series for c_4 and c_5.

Calculation of c-Functions, Continued

- Calculate c–functions of original argument by iteration of the quadruple–argument formulae:

  Let $c_0^{(N)}$, $c_1^{(N)}$, $c_2^{(N)}$, $c_3^{(N)}$, $c_4^{(N)}$, and $c_5^{(N)}$ denote c–functions calculated for reduced argument. Iterate quadruple–argument formulae

  $$c_5^{(k-1)} = [c_2c_3 + c_4 + c_5]^{(k)}/16$$

  $$c_4^{(k-1)} = [c_2c_2 + c_4 + c_4]^{(k)}/8$$

  $$c_3^{(k-1)} = [c_1c_2 + c_3^{(k)}/4 \qquad c_2^{(k-1)} = [c_1c_1]^{(k)}/2$$

  $$c_1^{(k-1)} = [c_0c_1]^{(k)} \qquad c_0^{(k-1)} = 2[c_0c_0]^{(k)} - 1$$

  for k = N, N−1, ... , 0. Note that if N = 0 to begin with, then iteration is not necessary.

Calculation of c-Functions, Summary Algorithm

a. Set N = 0 and set y = x, where x is input argument.
b. If |y| < 0.1 then go to Step d, else go to Step c.
c. Increment N by 1 & divide y by 4, then go to Step b.
d. Calculate

$$c_5 = (1-y(1-y(1-y(1-y(1-y(1-y/272)/210)/156)/110)/72)/42)/120;$$

$$c_4 = (1-y(1-y(1-y(1-y(1-y(1-y/240)/182)/132)/90)/56)/30)/24;$$

$$c_3 = 1/6 - yc_5;$$

$$c_2 = 1/2 - yc_4;$$

$$c_1 = 1 - yc_3;$$

$$c_0 = 1 - yc_2.$$

e. If N = 0 then exit, else go to Step f.
f. Decrement N by 1, calculate "new" c–functions of quadruple argument from "old" c–functions of previous argument, then go to Step e.

Summary of Presentation

- "Universal variables" is an algorithm that propagates the position and velocity of a secondary moving around its primary, and
  - the algorithm does not branch to three different sets of equations based upon the value of the eccentricity ($e < 1$, $e = 1$, or $e > 1$)

- Karl J. Stumpff published such a universal variables algorithm in 1947 using his c-functions

- Stumpff's c-functions generalize the sine and cosine functions
  - we now know how to calculate the first six: c_0, c_1, c_2, c_3, c_4, and c_5

- The solar system path of the recently discovered interstellar asteroid `Oumuamua is hyperbolic ($e > 1$), so:
  - we can choose to propagate its path using an algorithm that only works for hyperbolic paths, or
  - we can choose to use the more general "universal variables" equations that work for all conic paths

REFERENCES

[1] Stumpff, Karl J., Neue Formeln und Hilfstafeln zur Ephemeridenrechnung, *Astronomische Nachrichten,* Vol. 275 (1947), pp. 108-128.

[2] Mansfield, Roger L., To*pics in Astrodynamics,* Astronomical Data Service (Colorado Springs, 2003), pp. 267-268.

[3] Battin, Richard H., *Astronautical Guidance,* McGraw-Hill (New York, 1964), Section 2.8.

[4] Goodyear, William H., Completely General Closed-Form Solution for the Coordinates and Partial Derivatives of the Two-Body Problem, *Astronomical Journal,* Vol. 70, No. 3 (April 1965), pp. 189-192.

[5] Herrick, Samuel, Universal Variables, *Astronomical Journal,* Vol. 70, No. 4 (May 1965), pp. 309-315.

[6] Stiefel, E. L. and G. Scheifele, *Linear and Regular Celestial Mechanics,* Springer-Verlag (New York, 1971), Section 11 (see p. 43).