Kepler's Equation

This document describes several MATLAB functions that can be used to solve Kepler's equation for circular, elliptical, parabolic and hyperbolic orbits.

kepler1.m – Danby's method

This MATLAB function is based on a numerical solution of Kepler's equation devised by Professor J. M. A. Danby at North Carolina State University. Additional information about this algorithm can be found in "The Solution of Kepler's Equation", *Celestial Mechanics*, **31** (1983) 95-107, 317-328 and **40** (1987) 303-312.

The initial guess for Danby's method is

$$E_0 = M + 0.85\,\mathrm{sign}(\sin M)\,e$$

The fundamental equation we want to solve is

$$f(E) = E - e\sin E - M = 0$$

which has the first three derivatives given by

$$f'(E) = 1 - e\cos E$$

$$f''(E) = e\sin E$$

$$f'''(E) = e\cos E$$

The iteration for an updated eccentric anomaly based on a current value E_n is given by the next four equations:

$$\Delta(E_n) = -\frac{f}{f'}$$

$$\Delta^*(E_n) = -\frac{f}{f' + \frac{1}{2}\Delta f''}$$

$$\Delta_n(E_n) = -\frac{f}{f' + \frac{1}{2}\Delta f'' + \frac{1}{6}\Delta^{*2} f'''}$$

$$E_{n+1} = E_n + \Delta_n$$

This algorithm provides quartic convergence of Kepler's equation. This process is repeated until the following convergence test involving the fundamental equation is satisfied:

$$\left| f(E) \right| \le \varepsilon$$

where ε is the convergence tolerance. This tolerance is hardwired in the software to ε = 1.0e-10. Finally, the true anomaly can be calculated with the following two equations

$$\sin\theta = \sqrt{1-e^2}\,\sin E$$

$$\cos\theta = \cos E - e$$

and the four quadrant inverse tangent given by

$$\theta = \tan^{-1}\left(\sin\theta, \cos\theta\right)$$

If the orbit is hyperbolic the initial guess is

$$H_0 = \log\left(\frac{2M}{e} + 1.8\right)$$

where H_0 is the hyperbolic anomaly. The fundamental equation and first three derivatives for this case are as follows:

$$f\left(H\right) = e\sinh H - H - M$$

$$f'\left(H\right) = e\cosh H - 1$$

$$f''\left(H\right) = e\sinh H$$

$$f'''\left(H\right) = e\cosh H$$

Otherwise, the iteration loop which calculates Δ, Δ^*, and so forth is the same. The true anomaly for hyperbolic orbits is determined with this next set of equations:

$$\sin\theta = \sqrt{e^2 - 1}\sinh H \qquad \cos\theta = e - \cosh H$$

The true anomaly is then determined from a four quadrant inverse tangent evaluation of these two equations.

The syntax of this MATLAB function is

```
function [eanom, tanom] = kepler1 (manom, ecc)

% solve Kepler's equation for circular,
% elliptic and hyperbolic orbits

% Danby's method

% input

%  manom = mean anomaly (radians)
%  ecc   = orbital eccentricity (non-dimensional)
```

```
% output

%  eanom = eccentric anomaly (radians)
%  tanom = true anomaly (radians)
```

kepler2.m – Danby's method with Mikkola's initial guess

This function solves the elliptic and hyperbolic form of Kepler's equation using Danby's method and Mikkola's initial guess. This method uses a cubic approximation of Kepler's equation for the initial guess. Additional information about this initial guess can be found in "A Cubic Approximation For Kepler's Equation", *Celestial Mechanics*, **40** (1987) 329-334.

The elliptic orbit initial guess for this method is given by the expression

$$E_0 = M + e\left(3s - 4s^3\right)$$

where

$$s = s_1 + ds \qquad s_1 = z - \frac{\alpha}{2} \qquad ds = -\frac{0.078 s_1^5}{1+e}$$

and

$$z = \operatorname{sign}(\beta)\left(|\beta| + \sqrt{\alpha^2 + \beta^2}\right)^{1/3} \qquad \alpha = \frac{1-e}{4e + \frac{1}{2}} \qquad \beta = \frac{\frac{1}{2}M}{4e + \frac{1}{2}}$$

Updates to the eccentric anomaly are calculated using the same set of equations as those in Danby's method. For hyperbolic orbits this method uses the following for the correction to s

$$ds = \frac{0.071 s^5}{e\left(1 + 0.45 s^2\right)\left(1 + 4s^2\right)}$$

and the following initial guess for the hyperbolic anomaly:

$$H_0 = 3\log\left(s + \sqrt{1 + s^2}\right)$$

The syntax of this MATLAB function is

```
function [eanom, tanom] = kepler2 (manom, ecc)

% solve Kepler's equation for circular,
% elliptic and hyperbolic orbits

% Danby's method with Mikkola's initial guess

% input

%  manom = mean anomaly (radians)
```

```
%  ecc   = orbital eccentricity (non-dimensional)

% output

%  eanom = eccentric anomaly (radians)
%  tanom = true anomaly (radians)
```

kepler3.m – Gooding's two iteration method

This MATLAB function solves the elliptic form of Kepler's equation using Gooding's two iteration method. This algorithm performs two, and only two iterations when solving Kepler's equation. Additional information about this technique can be found in "Procedures For Solving Kepler's Equation", *Celestial Mechanics*, **38** (1986) 307-334.

The syntax of this MATLAB function is

```
function [eanom, tanom] = kepler3 (manom, ecc)

% solve Kepler's equation for elliptic orbits

% Gooding's two iteration method

% input

%  manom = mean anomaly (radians)
%  ecc   = orbital eccentricity (non-dimensional)

% output

%  eanom = eccentric anomaly (radians)
%  tanom = true anomaly (radians)
```

kepler4.m – parabolic and near-parabolic orbits

This MATLAB function solves Kepler's equation for heliocentric parabolic and near-parabolic orbits. It is based on the numerical method described in Chapter 4 of *Astronomy on the Personal Computer* by Oliver Montenbruck and Thomas Pfleger. This algorithm uses a modified form of Barker's equation and Stumpff functions to solve this problem.

The form of Kepler's equation solved by this function is

$$E(t) - e\sin E(t) = \sqrt{\frac{\mu}{a^3}}\left(t - t_0\right)$$

where

$E =$ eccentric anomaly

$e =$ orbital eccentricity

$\mu =$ gravitational constant of the Sun

$a =$ semimajor axis

$t =$ time

$t_0 =$ time of perihelion passage

The relationship between semimajor axis *a* and perihelion radius *q* is as follows:

$$a = \frac{q}{1-e}$$

By introducing the variable

$$U = \sqrt{\frac{3ec_3(E)}{1-e}}E$$

Kepler's equation is now given by

$$U + \frac{1}{3}U^3 = \sqrt{6ec_3(E)}\sqrt{\frac{\mu}{2q^3}}(t-t_0)$$

where $c_3(E) = (E - e\sin E)/E^3$. The `kepler4` function iteratively solves for *U*.

The heliocentric distance is determined from the expression

$$r = q\left(1 + \left[\frac{2c_2}{6c_3}\right]U^2\right)$$

The true anomaly is determined from the *x* and *y* components of the heliocentric position vector as follows:

$$\theta = \tan^{-1}(y, x)$$

where

$$x = q\left(1 - \left[\frac{2c_2}{6ec_3}\right]U^2\right)$$

$$y = 2q\sqrt{\frac{1+e}{2e}\left[\frac{1}{6c_3}\right]}c_1 U$$

The true anomaly can also be determined from

$$\tan\left(\frac{\theta}{2}\right) = \left(\sqrt{\frac{1+e}{3ec_3}}\frac{c_2}{c_1}\right)U$$

The *c* functions used in these equations are called Stumpff functions. They are named after the German astronomer Karl Stumpff and defined by the series

$$c_n(x) = \sum_{k=0}^{\infty}\frac{(-1)^k x^k}{(2k+n)!} \qquad k = 0,1,2,\ldots$$

For x real and $x \neq 0$, the first few terms are given by the following expressions:

$$c_0\left(x^2\right) = \cos x \qquad c_0\left(-x^2\right) = \cosh x$$

$$c_1\left(x^2\right) = \frac{\sin x}{x} \qquad c_1\left(-x^2\right) = \frac{\sinh x}{x}$$

$$c_2\left(x^2\right) = \frac{1-\cos x}{x} \qquad c_2\left(-x^2\right) = \frac{\cosh x - 1}{x}$$

The Stumpff functions also satisfy the recursion relationship defined by

$$x c_{k+2}\left(x\right) = \frac{1}{k!} - c_k\left(x\right) \qquad k = 0,1,2,\ldots$$

For $x = 0$,

$$c_n\left(x\right) = \frac{1}{n!}$$

It is most efficient to compute c_2 and c_3 by series and then compute c_0 and c_1 by recursion according to the following:

$$c_0\left(x\right) = 1 - x c_2$$

$$c_1\left(x\right) = 1 - x c_3$$

The syntax of this MATLAB function is

```
function [r, tanom] = kepler4(t, q, ecc)

% Kepler's equation for heliocentric
% parabolic and near-parabolic orbits

% input

%  t   = time relative to perihelion passage (days)
%  q   = perihelion radius (AU)
%  ecc = orbital eccentricity (non-dimensional)

% output

%  r     = heliocentric distance (AU)
%  tanom = true anomaly (radians)
```