Solving Kepler's Equation

M. Atakan Gürkan

1 The problem

Solving Kepler's equation for arbitrary epoch and eccentricity is a common problem in celestial mechanics. Below, I present my implementation of such a solver, written in `C`. It is an extension of Shepperd's (1985) work, and uses ideas from Danby (1992) and Mikkola (1999).

Given initial position and velocity and the time interval, the code calculates the new position and the velocity of a particle with the Hamiltonian

$$\mathcal{H} = \frac{p^2}{2m} - \frac{\mu}{r} - \frac{B_2}{r^2}\,. \tag{1}$$

2 Why a new routine

There are of course codes already available for this purpose; perhaps most notably the `SWIFT` routines[1]. They are heavily tested and *very* reliable. However, I wanted to write my own routines, for a number of reasons.

- It was not straightforward for me to reach machine accuracy with `SWIFT` routines. In particular, on my laptop (AMD, x86-64) with GNU Fortran compiler, I explicitly had to use "`-mfpmath=387`" flag. Unless this flag is used, the code generated uses the double precision SSE instruction set. Unlike the 387 coprocessor instructions, SSE instructions do not store the temporary results in 80 bit precision. Seemingly this causes a small error (this was pointed out to me by Patricia Verrier). However, the error is consistent, so it adds up linearly in time, and decreasing the timestep makes things worse.
  I probably did not need machine precision in the first place, but it provides a peace of mind.
- I wanted something compact and in `C`, so I could play around with it more easily, parallelize the code, port it to GPUs etc.
- I wanted to see if continued fractions led to faster code (The speed improvement I obtained was negligible).

[1] See: `http://www.boulder.swri.edu/~hal/swift.html`

- Continued fractions are intriguing on their own. The functions we need to solve Kepler's equation have surprisingly simple continued fraction expansions. Continued fractions allow infinite precision arithmetic (Gosper, 1972). Well, I do not know what to do about it at the moment, but it looks very interesting.

Please note that my experience with `SWIFT` routines is very limited. It is possible that one can make a few modifications to increase their accuracy while maintaining their reliability. I'd appreciate any feedback on this matter.

3 Solving Kepler's equation

For simplicity let us initially assume $B_2 = 0$. Then the orbit is a Keplerian conic section. We start by calculating the quantities

$$r_0 = |\mathbf{r}_0|, \quad v_0 = |\mathbf{v}_0|, \quad \sigma_0 = \mathbf{r}_0 \cdot \mathbf{v}_0, \quad \beta = \frac{2\mu}{r_0} - v_0^2\,, \tag{2}$$

where $\mathbf{r}_0$ and $\mathbf{v}_0$ are the initial position and velocity vectors, respectively. If $\beta > 0$, we have an elliptical orbit and can calculate the period

$$P = 2\pi\mu\beta^{-3/2}\,, \tag{3}$$

and the quantity

$$\delta U = n 2\pi\beta^{-5/2}, \quad n = \left\lfloor \frac{\delta t + P/2 - 2\sigma_0/\beta}{P} \right\rfloor\,, \tag{4}$$

where δt is the given time interval. This will be useful if $\delta t > P$, a situation that I do not encounter since I always choose timesteps much shorter than the period.

For our independent variable, we choose the initial value $u = 0$. The main loop consists of the following steps (prime denotes differentiation with respect to u):

$$q = \frac{\beta u^2}{1+\beta u^2}\,, \tag{5}$$

$$q' = \frac{2\beta u}{(1+\beta u^2)^2}\,, \tag{6}$$

$$q'' = \frac{2\beta}{(1+\beta u^2)^2} - \frac{8\beta^2 u^2}{(1+\beta u^2)^3}\,, \tag{7}$$

$$\tilde{U_0} = 1 - 2q\,, \tag{8}$$

$$\tilde{U_1} = 2(1-q)u\,, \tag{9}$$

$$U = \frac{16}{15}\tilde{U_1}^5 G_5(q) + \delta U\,, \tag{10}$$

$$U_0 = 2\tilde{U_0}^2 - 1\,, \tag{11}$$

$$U_1 = 2\tilde{U_0}\tilde{U_1}\,, \tag{12}$$

$$U_2 = 2\tilde{U_1}^2\,, \tag{13}$$

$$U_3 = \beta U + U_1 U_2/3\,, \tag{14}$$

$$r = r_0 U_0 + \sigma_0 U_1 + \mu U_2\,, \tag{15}$$

$$r' = 4(1-q)(\sigma_0 U_0 + (\mu - \beta r_0)U_1)\,, \tag{16}$$

$$r'' = -4q'(\sigma_0 U_0 + (\mu - \beta r_0)U_1) + 16(1-q)^2(-\beta\sigma_0 U_1 + (\mu - \beta r_0)U_0)\,, \tag{17}$$

$$\Delta t = r_0 U_1 + \sigma_0 U_2 + \mu U_3\,, \tag{18}$$

$$f = \Delta t - \delta t\,, \tag{19}$$

$$f' = 4(1-q)r\,, \tag{20}$$

$$f'' = 4(r'(1-q) - rq')\,, \tag{21}$$

$$f''' = -8r'q' - 4rq'' + 4(1-q)r''\,, \tag{22}$$

$$\delta u_1 = -f/f'\,, \tag{23}$$

$$\delta u_2 = -f/(f' + \delta u_1 f''/2)\,, \tag{24}$$

$$\delta u_3 = -f/(f' + \delta u_2 f''/2 + \delta u_2^2 f'''/6)\,, \tag{25}$$

$$u = u + \delta u_3\,. \tag{26}$$

Once the increment δu_3 or f is smaller than a preset tolerance, we exit the loop. The function $G_5(x)$ is a special case of Gauss's hypergeometric function: $G_5(x) = {}_2F_1(5,1;7/2;x)$ (Abramowitz and Stegun, 1972, Ch. 15). The position and velocities are calculated by

$$f = 1 - (\mu/r_0)U_2\,, \tag{27}$$
$$g = r_0 U_1 + \sigma_0 U_2\,, \tag{28}$$
$$F = -\mu U_1/(r r_0)\,, \tag{29}$$
$$G = 1 - (\mu/r)U_2\,, \tag{30}$$
$$\mathbf{r} = f\mathbf{r}_0 + g\mathbf{v}_0\,, \tag{31}$$
$$\mathbf{v} = F\mathbf{r}_0 + G\mathbf{v}_0\,. \tag{32}$$

This method of solution is almost identical to Shepperd's. The only thing I did is to increase the order of the iteration by using Danby's technique. In the actual implementation, if $q > 1/2$ during iteration, or if convergence is not achieved after 12 iterations, we stop, and try to cover δt in two steps of $\delta t/2$.

When $B_2 \neq 0$, we need two modifications. In the following, I adopt the approach of Mikkola (1999); see that paper for generating functions etc. First note that in the plane of motion, we can transform to polar coordinates (r, θ) and write the Hamiltonian as

$$\begin{aligned}\mathcal{H} &= \frac{1}{2}\left(p_r^2 + \frac{p_\theta^2}{r^2}\right) - \frac{\mu}{r} - \frac{B_2}{r^2} \\ &= \frac{1}{2}\left(p_r^2 + \frac{p_\theta^2 - 2B_2}{r^2}\right) - \frac{\mu}{r} \\ &= \frac{1}{2}\left(p_r^2 + \frac{p_\psi^2}{r^2}\right) - \frac{\mu}{r}\,,\end{aligned} \tag{33}$$

which is in Keplerian form. The first modification is

$$\beta = \frac{2\mu}{r_0} - v_0^2 + \frac{2B_2}{r_0^2}\,. \tag{34}$$

Furthermore, we cannot use f-g formulation directly, so at the end of the loop, we revert to a more general method, which is applicable for any motion with a radial force:

$$
\begin{aligned}
\mathbf{p}_\theta &= \mathbf{r}_0 \times \mathbf{v}_0\,, && (35)\\
p_\psi^2 &= p_\theta^2 - 2B_2\,, && (36)\\
f &= 1 - (\mu/r_0)U_2\,, && (37)\\
g &= r_0 U_1 + \sigma_0 U_2\,, && (38)\\
\xi &= \frac{g}{r_0^2 f + \sigma_0 g}\,, && (39)\\
y_\psi &= \xi \mathrm{at}_1(p_\psi^2 \xi^2)\,, && (40)\\
\dot{r} &= (\sigma_0 U_0 + (\mu - \beta r_0)U_1)/r\,, && (41)\\
\theta^2 &= p_\theta^2 y_\psi^2\,, && (42)\\
\mathbf{r} &= \frac{r}{r_0}(c_0(\theta^2)\mathbf{r}_0 + y_\psi c_1(\theta^2)\mathbf{p}_\theta \times \mathbf{r}_0)\,, && (43)\\
\mathbf{v} &= \frac{\dot{r}}{r}\mathbf{r} + \frac{1}{r^2}\mathbf{p}_\theta \times \mathbf{r}\,. && (44)
\end{aligned}
$$

In this formulation $c_0(x^2)$ and $c_1(x^2)$ are Stumpff functions, and $\mathrm{at}_1(x^2)$ is $\arctan(x)/x$. All these functions, and the function $G_5(x)$ mentioned earlier, have simple continued fraction expansions. However, in my implementation, I only calculate $G_5(x)$ by continued fraction expansion to keep things simple. An efficient way to do this is given by (Shepperd, 1985). For calculating c_0 and c_1 by continued fractions, probably the best way is to use the approach of Flanders and Frame (1987). The continued fraction for $\mathrm{at}_1(x)$ goes back to Lambert (1770) and Lagrange (1776) according to Olds (1963, Appendix II, formula 14):

$$
\mathrm{at}_1(x) = \frac{\arctan(\sqrt{x})}{\sqrt{x}} = \cfrac{1}{1 + \cfrac{1 \cdot x}{3 + \cfrac{4 \cdot x}{5 + \cfrac{9 \cdot x}{7 + \cfrac{16 \cdot x}{9 + \cfrac{25 \cdot x}{\ddots}}}}}}\,. \qquad (45)
$$

References

M. Abramowitz and I. A. Stegun. *Handbook of Mathematical Functions*. Dover, New York, 1972.

J. M. A. Danby. *Fundamentals of celestial mechanics, 2nd ed.* Willman-Bell, Richmond, 1992.

Harley Flanders and J. Sutherland Frame. Algorithm of the bi-month: Elementary transcendental functions. *The College Mathematics Journal*, 18(5):417–421, 1987. ISSN 07468342. URL http://www.jstor.org/stable/2686970.

R. W. Gosper. Continued Fractions. *MIT HAKMEM Artificial Intelligence Memo*, no 239:item 101, 1972. http://www.inwap.com/pdp10/hbaker/hakmem/cf.html and http://keithbriggs.info/cfup.htm.

S. Mikkola. Efficient Symplectic Integration of Satellite Orbits. *Celestial Mechanics and Dynamical Astronomy*, 74:275–285, August 1999. doi: 10.1023/A:1008398121638.

Carl Douglas Olds. *Continued Fractions*. Random House, 1963.

S. W. Shepperd. Universal Keplerian state transition matrix. *Celestial Mechanics*, 35:129–144, February 1985.