Orbits & Simulation

Part 4

Previously we looked at the Euler and Leapfrog methods of numerical integration and determined that they are not accurate enough to support serious study of orbits. This time we will look at the Hermite 4th Order method which gives us more reliable foundation for orbit simulation.

Hermite 4th Order Method

Snap, Crackle and Pop!

In the first part of these notes we introduced the concept of 'rate of change' or 'differential with respect to time'. In particular equation 1.10:

$$\mathrm{a} \equiv \dot{\mathrm{v}} \equiv \ddot{\mathrm{r}} \equiv \{\ddot{x}, \ddot{y}, \ddot{z}\}$$

"Acceleration is the rate of change of velocity and velocity is the rate of change of position".

In the Leapfrog method we tried to take into account the rate of change of acceleration by seeing how much the acceleration changed during the time step $(\mathbf{a}_{i+1} - \mathbf{a}_i)/\boldsymbol{\delta t}$. We can continue this approach indefinitely: whatever value we are using it will have a rate of change and the rate of change will have a rate of change etc. Some of these rates of change we already know:

$\mathbf{r}$ position
$\dot{\mathbf{r}}$ velocity = rate of change of position
$\ddot{\mathbf{r}}$ acceleration = rate of change of velocity

Some others have been given unofficial names:

$\dddot{\mathbf{r}} = \mathrm{j}$ jerk = rate of change of acceleration
$\ddddot{\mathbf{r}} = \mathrm{s}$ snap = rate of change of jerk
$\mathbf{r}^{(5)} = \mathrm{c}$ crackle = rate of change of snap
$\mathbf{r}^{(6)} = \mathrm{p}$ pop = rate of change of crackle

The Taylor Series

In 1715 the mathematician Brook Taylor showed that problems like *n*-body integration can be expressed as a series of the form:

$$\ddot{\boldsymbol{r}} = f(\boldsymbol{r})$$

And $$\mathbf{r}_{i+1} = \mathbf{r}_i + \dot{\mathbf{r}}_i\boldsymbol{\delta t} + \frac{1}{2!}\ddot{\mathbf{r}}_i\boldsymbol{\delta t^2} + \frac{1}{3!}\dddot{\mathbf{r}}_i\boldsymbol{\delta t^3} + \frac{1}{4!}\ddddot{\mathbf{r}}_i\boldsymbol{\delta t^4} + \cdots \frac{1}{m!}\overset{m}{\mathbf{r}}\boldsymbol{\delta t^m}$$

What this says is:

1. Acceleration is a function of position. This function we already know and called it “calculate accelerations” in equation 2.6.
2. The next position is based on an ever decreasing proportion of ever higher level differentials of the equations of motion.

We have used the first three terms of the Taylor series in the Leapfrog method. We also applied an estimate of the 4th term (jerk) by using the average acceleration $\frac{1}{2}(\mathbf{a}_i+\mathbf{a}_{i+1})$.

However, the correct value of jerk can be found by differentiation of the acceleration equation (2.6). (I am not going to attempt to do that!). However, if…

(2.6) $$\mathbf{a}_p = G\sum_{\substack{q=1,\\ q\neq p}}^{q=n} \frac{m_q}{r_{pq}^3}\mathbf{r}_{pq}$$

Then 4.1 $$\mathbf{j}_p \equiv \dot{\mathbf{a}}_p = G\sum_{\substack{q=1,\\ q\neq p}}^{q=n} m_q\left(\frac{\mathbf{v}_{qp}}{r_{qp}^3} - \frac{3(\mathbf{r}_{qp}\cdot\mathbf{v}_{qp})\mathbf{r}_{qp}}{r_{qp}^5}\right)$$

Note the expression $\mathbf{r}\cdot\mathbf{v}$ is the “dot product” of the two vectors and is given by:

$$\mathbf{r}\cdot\mathbf{v} \equiv \{r_x, r_y, r_z\}\cdot\{v_x, v_y, v_z\} = r_x v_x + r_y v_y + r_z v_z$$

(I have not discovered the proof of how equation 4.1 is derived from equation 2.6, but it is used in Aarseth & Makino 1992 and Hut & Makino 2007. Any experts in vector differentiation to the rescue?)

Hermite approach

We now have methods to calculate the first 4 terms of the Taylor series and we can make an estimate of the 5th (snap) by using the expression $(\mathbf{j}_i - \mathbf{j}_{i+1})/\boldsymbol{\delta t}$ as an estimated rate-of-change of jerk.

For each timestep of the integration, the Hermite 4th Order method uses the Taylor series, up to and including jerk, to calculate predicted next positions and velocities for each object (shown as $\mathbf{r}_{i'}$ and $\mathbf{v}_{i'}$ in the procedure below).

These predicted next positions and velocities are then used to calculate predicted new acceleration and jerk values $\mathbf{a}_{i'}$ and $\mathbf{j}_{i'}$.

Finally, 4th Order corrected values of position and velocity $\mathbf{r}_{i+1}$ and $\mathbf{v}_{i+1}$ are calculated employing revised estimates of average acceleration and jerk. (The full derivation of the expression for the corrected values can be found in Hut & Makino 2007.)

Note that, strictly speaking, the accelerations and jerks for the next step should be recalculated again after the 4th Order correction but this is not done in the Hermite scheme as the increase in accuracy is generally not considered worth the additional processing.

The integration steps for the Hermite method are:

1. Using current position $\mathbf{r}_i$ and velocity $\mathbf{v}_i$, calculate acceleration $\mathbf{a}_i$ and jerk $\mathbf{j}_i$ using equations 2.6 and 3.1.

2. Calculate provisional next positions and velocities:

$$\mathbf{r}_{i'} = \mathbf{r}_i + \mathbf{v}_i\delta t + \frac{1}{2}\mathbf{a}_i\delta t^2 + \frac{1}{6}\mathbf{j}_i\delta t^3$$

$$\mathbf{v}_{i'} = \mathbf{v}_i + \mathbf{a}_i\delta t + \frac{1}{2}\mathbf{j}_i\delta t^2$$

3. Recalculate predicted next acceleration $\mathbf{a}_{i'}$ and jerk $\mathbf{j}_{i'}$ based on the new positions $\mathbf{r}_{i'}$ and $\mathbf{v}_{i'}$ using equations 2.6 and 3.1.

4.1 Calculated corrected new position and velocity:

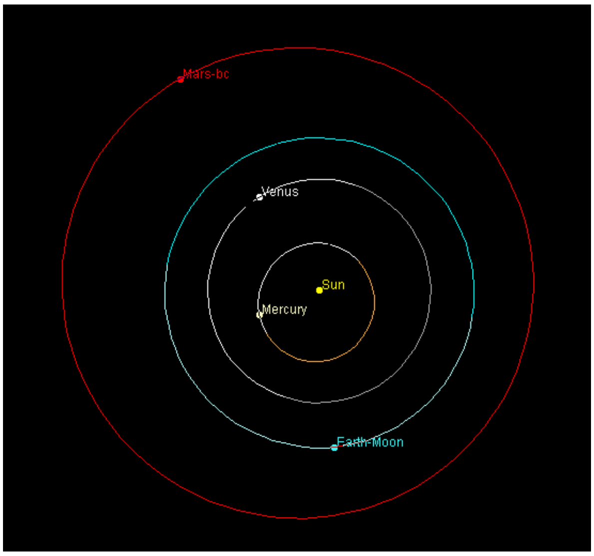
$$\mathbf{v}_{i+1} = \mathbf{v}_i + \frac{1}{2}(\mathbf{a}_i + \mathbf{a}_{i'})\delta t + \frac{1}{12}(\mathbf{j}_i - \mathbf{j}_{i'})\delta t^2$$

$$\mathbf{r}_{i+1} = \mathbf{r}_i + \frac{1}{2}(\mathbf{v}_i + \mathbf{v}_{i+1})\delta t + \frac{1}{12}(\mathbf{a}_i - \mathbf{a}_{i'})\delta t^2$$

All the above is done for all the bodies and for each iteration.

Results of Hermite 4th Order

Hermite integration after 1000 years, $\boldsymbol{\delta t}$ =0.1 day.



With a time step of 1 day we see that all the planets appear to have quite stable orbits and the integration is running fast. After 100 years of integration the energy error is similar to that produced by Leapfrog. Mercury's orbit is still precessing, but only slowly.

Using a time step of 0.1 day we see that the energy error is down to 2 parts in 10^{11} and there is no visible sign of

Mercury's orbit precessing. This is good performance and shows that the Hermite 4th order method is a good integrator when the time step is small enough.

On an intel i5 CPU this simulation will run the Solar System at about 5 years per second. The gif attached to the forum post shows the integration running at about this speed. The orbits of the inner planets can be seen to wobble in response to the perturbations of (mainly) Jupiter. The positions of the individual planets in their orbits are meaningless and due to the "strobe" effect of caturing a gif at 10 frames per second where the display is refreshing more often.

To be continued…