Scientific Particle Simulation and Visualisation

D.P.Playne and K.A.Hawick

daniel.playne@gmail.com, k.a.hawick@massey.ac.nz
Institute of Information and Mathematical Sciences, Massey University, Albany, North Shore 102-904, Auckland

Massey University

Introduction

Classical Newtonian particle simulations can be utilised for modeling complex many-body systems such as astronomical bodies, chemical interactions and even fluid dynamics. We show that scientifically accurate simulations can be integrated numerically and visualised in a realistic manner using the Java^{TM} programming language and the JOGL interface to OpenGL.

Simulation Model

Particle simulators use Newton's Laws of Motion to calculate the motion of a number of particles. In our Model each particle is considered to be a sphere with a mass, radius, position and velocity. The purpose of the simulator is to calculate the motion of the particles. Due to the discrete nature of computers this motion must be calculated over a series of time-steps. The motion of these particles is described by Newton's Laws of Motion:

$$v = u + at \quad s = \frac{1}{2}(u+v)t \tag{1}$$
$$s = ut + at^2 \quad v^2 = u^2 + 2as \tag{2}$$
$$s = vt + \tfrac{1}{2}at^2 \tag{3}$$

where:
t = time elapsed, s = distance
u = initial velocity, v = final velocity
a = acceleration

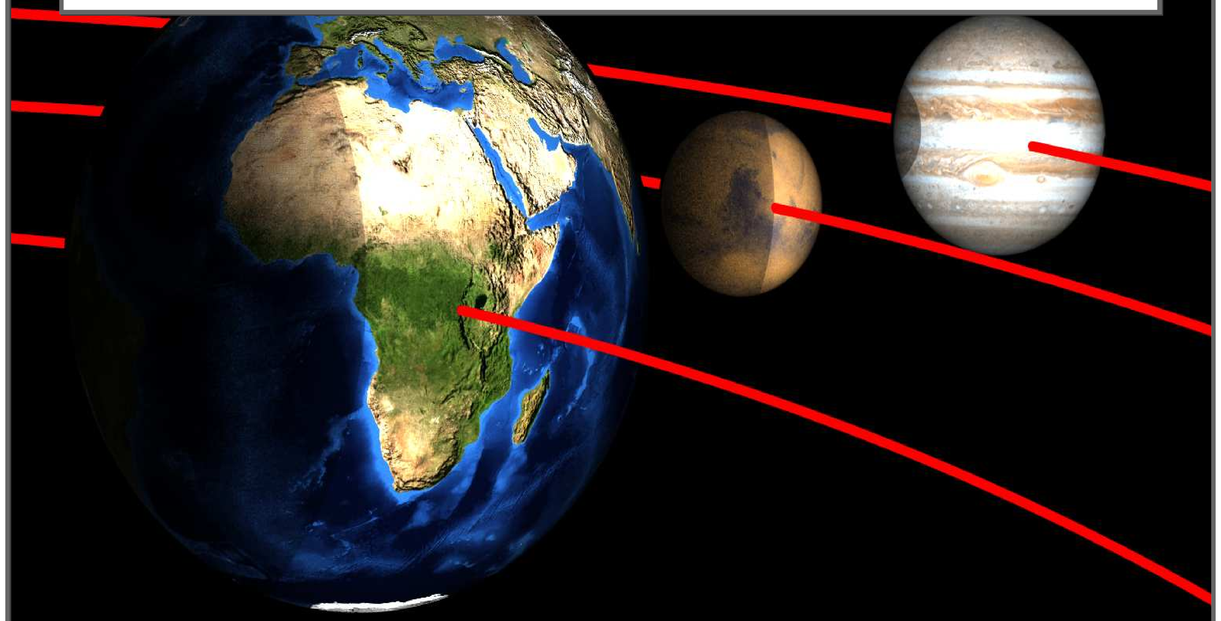
These simulators also often have a potential equation describing an attraction or repulsion between the particles. One commonly used potential energy equation is Newton's Law of Universal Gravitation as embodied as the force equation:

$$F = \frac{GM_j m_i}{r_{i,j}^2} \tag{4}$$

This potential equation describes the attraction due to gravity between the i, jpair of particles. The total potential of each particle is these potentials summed. In planetary motion simulators, each particle represents a planet in space. Newton's Laws of Motion are only capable of dealing with constant acceleration and the acceleration of each particle is continuously changing (as the particle moves the position-dependent force of gravity also changes). Because the change of acceleration is not constant, the motion of the particles must be integrated over time.

Simulation Visualisation

Visualisation is an important part of any simulation, it is the best way to convey the large amount of data available. The human visual system can easily process the images and easily recognise any recurring patterns or interesting events within the system. In general, particles can be visualised depending on their size and position; however, in some cases additional information can be visualised such as the texture which can make each individual particle easily identifiable (see background image for textured planets). The simulator/visualiser used to create this image utilises two separate rendering engines. The first is a real-time Z-buffer engine which creates a simple low-quality visualisation of the system. This engine was created using the JOGL library [1] which allows a Java^{TM} program to make direct calls to the OpenGL graphics library. The second engine is a slower ray-tracer which creates a high-quality video of a pre-recorded time sequence. This engine can be used to render an interesting interaction after the simulation has finished so that it can be re-examined in higher detail. This engine was implemented using the POVRay rendering library [2] and was used to create the background image shown below.



Particle Motion Integration

The simplest method of integrating the motion of the planets over time is Euler's method of integration. It approximates the acceleration and velocity of the particle to be constant over the entire time-step. This is effectively ignoring any change during the time-step and is a fundamentally unstable method (see Figure: 1). A far more stable and accurate method of integration is the Runge-Kutta 4th order method.

Stability Results

Implementing the Runge-Kutta 4th order method provides the stability and accuracy that is vital to a scientifically useful simulation. One method of testing the accuracy and stability of a simulation is to measure the energy of the system. This total energy can be found by calculating the kinetic and potential energy of each particle. The potential energy is defined by the potential equation of the simulation (in the case of gravity $-\frac{GMm}{r}$) and the kinetic energy can be calculated according to the formula $\frac{1}{2}mv^2$ where m and v are the particle's mass and velocity. The total energy of the system should remain constant throughout the simulation and so measuring it over time can give a good measure of the stability of the system. Figure: 1 shows a plot of the energy of the simulation calculated using the Euler method and the better Runge-Kutta 4th order method.

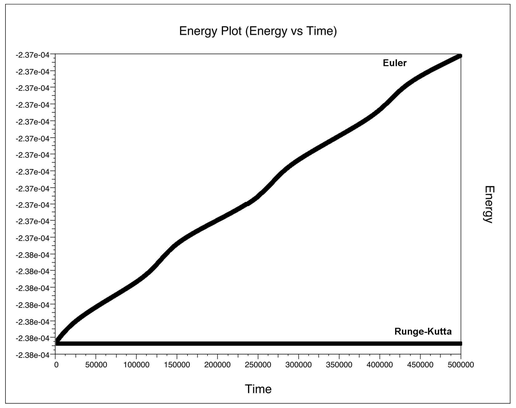


Figure 1: Plot of total energy over time showing cumulative error in the Euler method.

Runge-Kutta

The Runge-Kutta 4th order method integrates the motion of the particles by approximating the total acceleration and velocity of the particle at the n^{th} time-step according to several intermediate K values. This approximation is performed according to the following formulae [3]:

$$k_1 = f(t_n, y_n) \qquad k_2 = f(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_1) \tag{5}$$
$$k_3 = f(t_n + \tfrac{h}{2}, y_n + \tfrac{h}{2}k_2) \qquad k_4 = f(t_n + h, y_n + hk_3) \tag{6}$$
$$y_{n+h} = y_n + \tfrac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \tag{7}$$

where:
t_n is time of the simulation, y_n is position of the particle
$f(t, y)$ is the change (velocity and acceleration) of the particle.

Summary & Conclusions

Numerical methods can be used to calculate the motion of particles over a discrete time-step and higher-order integration methods can be used to provide a simulation with the numerical stability and accuracy vital for any scientific research. By calculating the energy of the system, this stability can be numerically measured to test the reliability of different algorithms. Numerically stable algorithms can be used to test a number of different scenarios. These simulations can be rendered in real-time and as high-quality video to allow a human observer to identify patterns and examine interesting phenomena within the simulation. This work combines both scientifically meaningful simulation and quality computer graphics. For more details see CSTN-057: Notes on Particle Simulation and Visualisation [4] at `www.massey.ac.nz/~kahawick/cstn/057/`.

We are currently using this planetary motion simulator to investigate a number of different methods of handling collisions. Current methods examined include combination, reflection and attachment.

References

[1] T. Bryson and K. Russell, "Java Binding for OpenGL (JOGL)," 2007.

[2] POV-Team, "Persistance of Vision Raytracer." http://www.povray.org/.

[3] W.H.Press, B.P.Flannery, S.A.Teukolsky, and W.T.Vetterling, *Numerical Recipes The Art of Scientific Computing*, ch. 17, pp. 907–910. Cambridge University Press, 3rd ed., 2007. Runge-Kutta Integration.

[4] D. P. Playne, "Notes on particle simulation and visualisation," Hons. Thesis, Computer Science, Massey University, June 2008.