Moonglum Simulation - User Manual

Augustus Porter
augustusjdporter@gmail.com

February 21, 2016

1 Introduction

The Moonglum project represents my attempt at making astrophysical simulations which are runnable on home PC technology. It is a code base providing the engine for N-Body simulations of astrophysical phenomena, with current functionality extending to Galaxy simulations and Planetary systems simulations.

2 Building the Project and Prerequisites

It is possible to download the zipped code repository from *https://github.com/augustusjdporter/Moonglum*. To compile, unzip the repository, enter the main *Moonglum* directory and type *make*; the binary executable *Moonglum* will be output.

In order to build, your computer will need to run a UNIX based operating system, *g++* compiler capable of compiling with the *-std=c++0x* flag. In order to make the plots with the included *python* scripts, your computer will need the *ipython* interpreter installed, along with the *matplotlib*, *numpy*, and *pylab* libraries.

3 Running a Simulation

To run the simulation, in the *Moonglum* directory, type:

./Moonglum [simulation name] [path to configuration file]

Setting the parameters of the simulation is completely done through xml-syntax configuration files - no changes need to be made to the executable file.

A directory with the input simulation name is made under the "galaxy-simulation/Coords" or "planetary-simulation/Coords" directory, depending whether the simulation is galactic or planetary, and is populated with the simulation output data. A detailed description of the output data is given in section 3.3.

3.1 The Configuration File

The configuration file is and xml-format file where the settings of the simulation are set - it is where you can define the astrophysical bodies and systems which are simulated. There are currently two strands of simulation which may be run by *Moonglum*, galactic or planetary, each with their own structure of configuration files.

The configuration files are read using the *rapidxml* tool. Though this is a very useful and easy to use tool, it does mean that if there is an error in the configuration file the program tends to crash with no explanation - I apologise in advance if this happens to you.

3.1.1 Planetary Configuration File

A planetary configuration file allows for a simulation of stars, planets and protoplanetary clouds. The structure of the configuration file is based around stars, and then defining planets and planetary clouds which orbit the stars. Figure 1 displays an example of a simple planetary configuration file.

```xml
<?xml version="1.0" encoding="utf-8"?>
<SimulationProfile>
    <simulationType>planetary</simulationType>
    <timestep>14400</timestep> <!--in seconds-->
    <numberOfSteps>10000</numberOfSteps>
    <samplingRate>12</samplingRate> <!--if 2 only takes snapshots every second step etc-->
    <Star name="Sun" mass="1" x="0" y="0" z="0" xVel="0" yVel="0" zVel="0" radius="1" logTrajectory="1">
        <Planet name="Earth" mass="1" inclination="0" orbitalRadius="1" orbitalPeriod="1" radius="1" logTrajectory="1"></Planet>
        <ProtoplanetaryCloud name="ProtoplanetaryCloud" numberOfPlanetesimals="11000" mass="0.08" xScale="4" yScale="4" zScale="0.2"></
        ProtoplanetaryCloud>
    </Star>
</SimulationProfile>
```

Figure 1: An example of a planetary simulation configuration file. This particular example can be found under *Moonglum/planetary-simulation/Configs/beerJournal.xml*.

Each of the "nodes" in the configuration file defines a parameter to be used in the simulation, and they affect the simulation as follows:

- *simulationType* - this defines the simulation as a planetary simulation, and is used to tell *Moonglum*'s xml-reader that it should read in the planetary config format.
- *timestep* - this defines the frame time in-between acceleration calculations and position updates in the simulation. The units of the timestep are in seconds. A smaller timestep will result in higher fidelity of the simulation, but will reduce the amount of simulated time which the simulation can cover in the same calculation time.
- *numberOfSteps* - this parameter defines the number of timesteps to calculate in the simulation.
- *samplingRate* - this parameter defines the rate at which simulation data should be recorded. It defines the number of timesteps between saving data, so a sampling rate of 2 would result in data being saved every second timestep etc.
- *Star* - this is where the root of the planetary simulation, the Star, is defined. It is possible to define any number of stars in the configuration file. Here you may give it a name, mass (in units of Solar Mass), x,y,z position in (units of AU), a velocity with respect to the static coordinates (in units of metres per second), radius (in units of Solar Radius), and a directive as to whether to log the Star's trajectory (1 = yes, 0 = no). Under the Star "node", you may define any number of planets or protoplanetary clouds to orbits the Star.
- *Planet* - This defines a planet which orbits a star. In order for successful parsing of the config file the planet node must be within the star node. Here you may define the name of the planet, mass (in units of Earth masses), inclination (in degrees), orbital radius (in AU), orbital period (in years), radius (in Earth radii), and whether to log its trajectory.
- *ProtoplanetaryCloud* - here is where you may define a protoplanetary cloud to orbit a star. The protoplanetary cloud node must be within the star node. You may define the number of planetesimals that make up the cloud, the mass of the entire cloud (in Solar Masses. The mass is divided up evenly between all of the planetesimals), and the x,y,z scale heights of the cloud (in AU).

3.1.2 Galaxy Configuration File

3.2 Python plotting script

3.3 Output Data

4 Future plans

- Extend the application to include cosmological simualtions.

- Comment the code (especially header files) so it is easier to understand from an outside view.
- Include Relativistic effects (both special and general).
- Move the computationally heavy calculations from the CPU to GPU to decrease computation time.