

Python N-Body Problem Solver

Barry Wang

1 PACKAGE INSTALLATION

Please “cd” into the project directory (N-Body) and run “install_packages.sh”. This script will install all necessary packages for this project and activate a virtual python environment at the current directory. Note that you need to manually deactivate this virtual environment if you wish to do so after inspecting this project.

2 PROJECT DESCRIPTION

This project uses the “numpy” package as its compute kernel and uses numerical integration to solve arbitrary N-Body problems. This project implements a “NBody.Solver” module in which a iterable “Solver” class is contained. The Solver object can load a simulation source and then solve the N-Body problem, optionally displaying or saving an animation of the simulation. It can also save the resulting data as a new simulation source for incremental simulation. For each iteration of the solver object, the solver updates its internal states (objects’ position, velocity, etc.). If visualization is enabled, the solver will update its internal state to the time when the next frame should be captured. Otherwise, it advances itself by 1 millisecond in the simulation.

To achieve an iteration of the solver object, many smaller compute “ticks” are performed. For each tick with a defined small time interval, the gravitational acceleration for each object resulting from every other object is computed using classical mechanical assumptions and then integrated to get their respective velocities and positions in the next time frame. All computations are vectorized for maximum performance. Numerous optimizations such as frame precomputing are implemented to facilitate simulation and animation rendering.

A simulation source is provided with a configuration ‘json’ file. A demonstration configuration file is included in the project directory. A demonstration application is also included, the usage of which can be accessed with “./demo.py -h”

This software is made public on GitHub on the day of submission. More detailed description on usage can be found at <https://github.com/Chutian-Wang/N-Body.git>

3 DEMONSTRATION AND RESULTS

Demonstration

To run the demonstration program, it is recommended to provide optional arguments so that an interesting animation can be viewed. A very simple two body system will be solved for 30 seconds but not displayed as an animation if no arguments are specified. Run “./demo.py -h” to view a help message which explains how this demonstration application is intended to be used. Alternatively, you may run “./demo.py -v -c demo_config.json -t 30 -s demo_result.json” which is the example usage that will be displayed by the help message. This command will source the demonstration configuration file, display an animation, and save the simulation result data. To run your own simulation, you can create a new configuration file and source this file when running the demonstration application and you may reference the included demonstration configuration file (“demo_config.json”) to create your own configuration file. Alternatively, you may create your own application by importing the “NBody.Solver” module.

Results

In general, the performance of this module is bottle-necked by the “matplotlib” auxiliary functions if visualization is enabled and small to medium problem sizes. A progress bar with simulation or frame generation speed in ms/s or frame/s is displayed when animation is not viewed directly.

A 6-body simulation animation (demo_video.gif) is included in the project directory. This configuration is tested to run at 12.5 simulation seconds per real time second on a RPi 5, 8GB version.