Hannu Pelttari

014158977

Planet simulation report

Introduction

Objective of this problem was to create a program that simulates planetary motion for N number of objects. N-body problems aren't viable to be solved analytically since the power series converge slowly. This means that the only practical solution is to solve them numerically.

Methods

The program uses velocity verlet algorithm to calculate trajectories for the planets. The velocity verlet algorithm which uses the current step to calculate the next particle positions, velocities and accelerations. Accelerations were calculated using Newton's universal law of gravitation where $F=G*(m_1*m_2)/r^2$. Since F=ma if the mass doesn't change the equation was divided with the mass of the object which acceleration is calculated to get the acceleration without first calculating the force.

Implementation

The program consists of three separate modules and a main program. One of the modules just has the gravitational constant defined so it is not that useful as a separate module for now but if the program is later expanded to solve different problems it might be handy so it will remain as a separate module. Then there are modules readvalues and velocityverlet. The readvalues module contains a subroutine that reads the initial values (masses, velocities, positions and the number of objects) from a user created text file. The velocityverlet module contains a subroutine that calculates the positions, velocities and accelerations of the objects using the velocity verlet algorithm. The main program calls the subroutine from readvalues, intialized a couple of variables, prints information about the simulation to the screen, calls the velocity verlet subroutine and writes the positions of the objects to a text file during the simulation.

The masses, positions and velocities of the planets are read to their own arrays. Velocities and positions in x,y and z directions are in their own arrays. The same holds for the accelerations. The first object in each of the arrays is the center object which is the reference frame of the simulation so no acceleration is calculated to that object and it's velocities and positions should be 0.

All of the modules and the main program should be first compiled to create .o -files. For example with linux commands "gfortran -c constanst.f90" to compile the constant module. This should be done similarly to the other files. Next step is to link all of the files together: gfortran constants.o readvalues.o velocityverlet.o simulation.o.

Results

The program gives fairly accurate results in the sense that the planet orbits for the solar system look like they should and the orbits are stable. The orbital periods aren’t that accurate however and the simulated orbital period of Jupiter is 271 days longer than the real orbital period days even if the time step is as short as 50 seconds. Time step of 20 seconds gives fairly accurate orbital period though which is just a few days longer than the real one. At 100 second time step the simulated orbital period is 119 days shorter than the real one. Even at 100000 second time step the accuracy isn’t that bad though. Efficiency could probably be a bit better also since it takes quite a while to simulate the whole solar system.

When simulating the whole solar system with 500 second time step the simulated orbital period for Mercury is about 90 days which is about two days longer than the actual orbital period. Simulated orbital period for Venus is 230 days which is about five days longer than the actual. For Earth the simulation gives 392 days and for Mars 700 days.

When approximating one year with the program with a 5 second time step it gives 346 days as a result. One month simulation gives 28 days as a result with the same time step

Overall the efficiency isn’t that bad though and the simulation of solar system has many more iterations than the simpler ones so it is expected to take a lot longer.

Conclusions

The program gives fairly accurate results for this purpose but if one wishes to get more accurate results then the calculation methods must be used. The velocity verlet agorithim could be replaced with a more accurate but more complex algorithm. The initial positions of the objects also might have some effect to the solar system simulation. The simulations were done with all of the planets lined up on the x-axis.