

ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ И  
ПРОГРАММИРОВАНИЕ

УДК 681.3: 528.91

**А. В. Аксенов** – студент кафедры вычислительных систем и сетей

**А. Ю. Козловский** – научный руководитель

**МОДЕЛИРОВАНИЕ СИСТЕМЫ ЗЕМЛЯ–ЛУНА–АСТЕРОИД**

Двести лет назад были открыты первые астероиды между Юпитером и Марсом. Позднее там был выделен главный пояс астероидов, насчитывающий около 98 % малых небесных тел. К двадцать первому веку их число превысило 400 тысяч. Предполагается, что в Солнечной системе может находиться от 1.1 до 1.9 миллиона объектов, имеющих размеры более 1 км. Наибольший научный интерес представляют астероиды, чьи орбиты проходят в опасной близости к Земле. Необходимо исследовать такие сближения, чтобы прогнозировать их возможные последствия, которые могут оказаться катастрофическими [1].

Сложность аналитического прогнозирования ситуации заключается в том, что невозможно рассчитать с достаточной точностью все условия и факторы, воздействующие на движение астероида. Так, прохождение вблизи Земли или Луны может изменить траекторию астероида, в результате чего в ходе следующего сближения возникает вероятность столкновения. Также существуют сложности с отслеживанием текущей траектории астероида [2]. Одним из решений проблемы оценки астероидной обстановки можно назвать моделирование ряда ситуаций с определенными диапазонами начальных параметров.

Моделирование технических процессов, а также создание моделей реальных объектов является актуальной задачей для решения широкого круга технических проблем. В большинстве случаев для моделирования сложных систем привлекаются компьютеры, возрастающие вычислительные мощности которых позволяют строить динамические модели, как аналитические, так и имитационные с высокой точностью и наглядностью, что немаловажно, как в научном, так и в образовательном плане. Компьютерные модели проще и удобнее исследовать, они позволяют проводить вычислительные эксперименты, реальная постановка которых затруднена или может дать непредсказуемый результат. Логичность и формализованность компьютерных моделей позволяет выявить основные факторы, определяющие свойства изучаемых объектов, исследовать отклик физической системы на изменения ее параметров и начальных условий.

Разработанная в рамках данного проекта программа моделирует систему трех небесных тел: Земли, Луны и астероида. Она позволяет исследовать поведение системы при различных начальных условиях, представляемых параметрами астероида: его массой, положением в пространстве, величиной скорости и ее направлением. Результатом работы программы является наглядная динамическая интерактивная 3D модель.

Модель вводит некоторые допущения, по сравнению с реальной системой.

1. Объекты, составляющие моделируемую систему, находятся в вакууме и не подвергаются никаким видам воздействий извне, поэтому систему можно считать замкнутой.

2. Единственный вид взаимодействия между объектами системы – гравитационное.

3. Размеры небесных тел, в отличие от их масс, не соответствуют действительности в интересах наглядности и прямо пропорциональны массам.

4. При малых по сравнению с расстоянием между телами перемещениях, происходящих за небольшой промежуток времени, сила взаимодействия тел считается постоянной. При таком условии можно легко найти положение тела в конце этого промежутка времени относительно его положения в начале. Поэтому вводится понятие коэффициента времени, задающего временной промежуток, за который тела совершают достаточно малые перемещения. Иными словами, моделирование осуществляется дискретно, то есть координаты объектов и силы взаимодействия между ними просчитываются через определенные, задаваемые пользователем, кванты времени.

По закону всемирного тяготения, два тела действуют друг на друга с силой, прямо пропорциональной произведению их масс и обратно пропорциональной квадрату расстояния между ними.

$$\vec{F} = \gamma \frac{m_1 m_2}{r^3} \vec{r}, \quad (1)$$

где  $\gamma$  – гравитационная постоянная,  $r$  – расстояние между телами. В проекции на ось X:

$$F_x = \gamma \frac{m_1 m_2}{(\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2})^3} (x_2 - x_1).$$

Здесь и далее для краткости будем рассматривать проекции только на ось X, в случае если проекции на другие оси аналогичны.

По второму закону Ньютона:

$$\vec{F} = m_1 \vec{a}. \quad (2)$$

В проекции:  $\ddot{x} = \frac{1}{m} F_x$ . Откуда:  $\ddot{x}_1 = \gamma \frac{m_2}{(\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2})^3} (x_2 - x_1)$ .

Для системы трех тел:

$$\ddot{x}_1 = \gamma \left( m_2 \frac{x_2 - x_1}{(\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2})^3} + m_3 \frac{x_3 - x_1}{(\sqrt{(x_3 - x_1)^2 + (y_3 - y_1)^2 + (z_3 - z_1)^2})^3} \right).$$

Таких дифференциальных уравнений в системе будет девять. Они будут описывать движение трех тел в трехмерном пространстве. Аналитическое решение представляет собой большую сложность [3].

В программе реализовано численное решение задачи трех тел явным методом Эйлера. В каждой следующей точке вычисляется новое значение величины, основываясь на знании ее значения в предыдущей точке, а также ее производной в предыдущей точке:  $y^{k+1} = y^k + hf(x^k, y^k)$ , где  $k$  – номер предыдущего шага,  $h$  – величина шага дискретизации,  $f(x^k, y^k)$  – правая часть уравнения, разрешенного относительно первой производной величины (тангенс угла наклона касательной). Таким образом, мы вычисляем скорости в каждой новой точке, основываясь на знании первой производной скорости по времени, то есть ускорения.

Явный метод Эйлера имеет первый порядок точности относительно шага дискретизации, и выбран исключительно в целях упрощения алгоритма.

Рассмотрим на примере взаимодействия двух тел численное решение, применяемое в программе.

По определению:

$$\vec{v} = \vec{a}t + \vec{v}_0. \quad (3)$$

Формула пути при равноускоренном движении:  $\vec{s} = \vec{v}_0 t + \frac{\vec{a}t^2}{2}$ .

$$\text{Тогда } \vec{s} = \frac{\vec{v}_0 t + \vec{v} t}{2} = \frac{\vec{v}_0 + \vec{v}}{2} t.$$

Из уравнений в векторной форме переходим к скалярным.  
Запишем в виде проекции уравнение (3):

$$v_{x1}^{k+1} = v_{x1}^k + a_{x1}^k t. \quad (4)$$

Поскольку начальная скорость для удобства задается в сферической системе координат, на первом шаге итерации перейдем от нее к декартовой, вычислив проекции вектора скорости на оси координат:

$$v_x^{k+1} = v^k \sin \varphi \cos \theta;$$

$$v_y^{k+1} = v^k \cos \varphi;$$

$$v_z^{k+1} = v^k \sin \varphi \sin \theta,$$

где  $\varphi$  – зенитный угол скорости,  $\theta$  – азимутальный угол скорости,  $k=0$ .

Формула для вычисления ускорения получается объединением уравнений (1) и (2) в скалярной форме:

$$a_{x1}^k = \gamma \frac{m_2(x_2^k - x_1^k)}{(\sqrt{(x_2^k - x_1^k)^2 + (y_2^k - y_1^k)^2 + (z_2^k - z_1^k)^2})^3}. \quad (5)$$

После объединения уравнений (4) и (5), имеем:

$$v_{x1}^{k+1} = v_{x1}^k + \gamma \frac{m_2(x_2^k - x_1^k)}{(\sqrt{(x_2^k - x_1^k)^2 + (y_2^k - y_1^k)^2 + (z_2^k - z_1^k)^2})^3} t. \quad (6)$$

Новая координата X будет определяться следующей формулой:

$$x_1^{k+1} = x_1^k + s_{x1}^k, \quad (7)$$

где  $s_{x1}^k$ , – проекция перемещения объекта за квант времени, определяемая из уравнения:

$$s_{x1}^{k+1} = \frac{v_{x1}^k + v_{x1}^{k+1}}{2} t. \quad (8)$$

Отсюда видно, что движение на участке между точками дискретизации является равноускоренным только формально. Фактически же тело движется равномерно со скоростью, равной среднему арифметическому скоростей на концах этого участка.

Объединяя зависимости (6)–(8), получаем итоговую рекуррентную формулу расчета координаты:

$$x_1^{k+1} = x_1^k + \frac{2v_{x1}^k + \gamma \frac{m_2(x_2^k - x_1^k)}{(\sqrt{(x_2^k - x_1^k)^2 + (y_2^k - y_1^k)^2 + (z_2^k - z_1^k)^2})^3} t}{2} t.$$

Программа написана на высокоуровневом интерпретируемом языке программирования Python версии 2.5. Его особенностями являются простота и удобочитаемость синтаксиса, относительная быстрота выполнения, наличие большого количества библиотек расширения (packages) и кроссплатформенность. Интерпретатор Python представлен в откомпилированном виде и в виде исходников для операционных систем Windows и Mac OS X на официальном сайте, а в большинстве дистрибутивов UNIX-подобных систем он включен изначально.

В качестве средства реализации графического интерфейса пользователя была выбрана библиотека Qt версии 4.3, которая лежит в основе графической среды UNIX-подобных систем KDE и выпускается, в том числе, и под лицензией GPL. Кроме родной для нее X11, она портирована на платформы Windows и Mac OS X. Помимо этого, к ее достоинствам можно отнести четко продуманную систему визуальных компонентов, чистый объектно-ориентированный дизайн, ин-

туитивность интерфейсов классов, что позволяет строить графический интерфейс, не прибегая к услугам редактора форм. Следует также отметить высокое быстродействие компилированного кода C++, на котором написана эта библиотека. Последнее особенно актуально, если учесть интерпретируемость, а, следовательно, и медлительность самого Python. Таким образом, мы снимаем с интерпретатора работу по организации ресурсоемкого по своей природе графического интерфейса. Для взаимодействия Python и Qt необходима интерфейсная библиотека PyQt.

Трехмерные построения выполнены с помощью библиотеки visual языка Python, которую можно найти в составе пакета VPython. Он также портирован на платформы Windows, X11 и Mac OS X. В данной работе использовалась версия VPython 2.5. Достоинством библиотеки является то, что она является объектно-ориентированной и высокоуровневой, что позволяет сконцентрировать внимание программиста на логике работы алгоритма, абстрагируясь от деталей реализации.

Таким образом, программный состав проекта обеспечивает удобочитаемость кода, возможность расширения и кроссплатформенность приложения.

Пользовательский интерфейс состоит из главного окна программы и окна трехмерного моделирования, которое вызывается нажатием на кнопку «Выполнить». В главном окне (рис. 1) можно установить начальные условия моделирования (параметры астероида, коэффициенты масштаба и времени) или выбрать один из предустановленных вариантов.

В окне моделирования (рис. 2) мы можем наблюдать движение объектов, их гравитационное взаимодействие, проявляющееся в изменении закона траектории движения тела, а также их визуализированные траектории. Манипулируя мышью, мы меняем ориентацию камеры, а также регулируем коэффициент приближения.

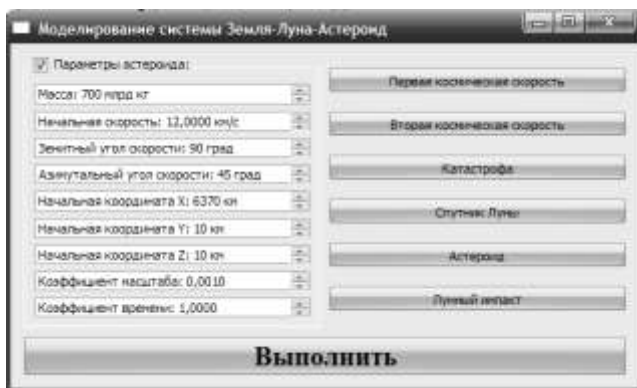


Рис. 1 – Главное окно программы

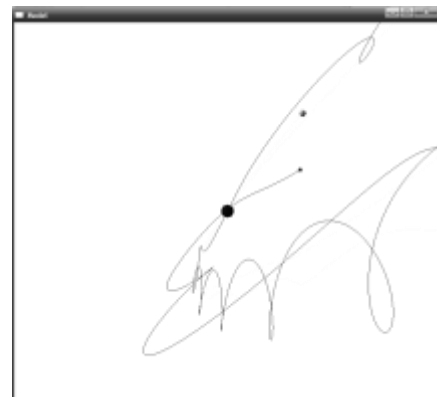


Рис. 2 – Окно моделирования

Сферы применения данной модели могут быть различны. Конечно, квантование времени порождает неточности и накладывает ограничение на ее использование в научных и исследовательских целях. Но можно с уверенностью говорить о практичности и целесообразности применения этой модели в сфере образования. Наглядная модель взаимодействия небесных тел может послужить хорошей иллюстрацией, к примеру, закона Всемирного тяготения или законов Кеплера. Также программа может быть использована для наглядного иллюстрирования поведения астероида в околоземном пространстве.

#### Библиографический список

1. <http://www.geokhi.ru/~meteorit/burba-lil.html>
2. <http://neo.jpl.nasa.gov/>
3. <http://faculty.ifmo.ru/butikov/Projects/CollectionR.html>