6. Эльконин Д. Б. Избранные психологические труды. - М: Педагогика, 1989.

Куканов М.А.

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ С ПОМОЩЬЮ VBA ФУНКЦИЙ MS EXCEL

matmrio@edurm.ru
Мордовский республиканский институт образования (МРИО)
г. Саранск

В последнее время все большее распространение получает использование офисного пакета *MS Excel* как средства учебного моделирования. Этому способствуют следующие обстоятельства: во-первых, широкая распространенность и относительная простота работы с ним, во-вторых, приемлемая ценовая доступность, которая выгодно отличает от специализированных пакетов *Mathcad, Matlab, Maple, Mathematica* и других подобных им [1], в третьих, наличие таких сервисных функций, как построение графиков, решение систем уравнений, нахождение минимумов и максимумов с дополнительными ограничениями, которые позволяет решать задачи линейного и нелинейного программирования. Однако, в основном, эти функции рассчитаны на моделирование в сфере экономики.

К большому сожалению, в составе библиотеки *MS Excel* нет функций решения дифференциальных уравнений и их систем, нет даже функции численного вычисления интегралов. Однако эти трудности вполне преодолимы благодаря наличию встроенного языка программирования *VBA (Visual Basic for Application)*. Этот язык является подмножеством популярного языка *MS Visual Basic*. С его помощью легко можно запрограммировать новые функции и включить их в пользовательскую библиотеку [2]. Далеко не все знают, что работа в среде *VBA MS Excel* намного удобнее, чем, например, в среде *Turbo Pascal, Delphi* и даже *MS Visual Basic*. Главное преимущество заключается в наличие готового интерфейса в виде электронных таблиц, которые можно использовать для хранения и анализа массивов числовых данных.

Нынешняя практика использования офисного пакета *MS Excel* напоминает ситуацию, когда при решении текстовых задач применяется арифметический подход, а не алгебраические методы. Подавляющее большинство примеров моделирования в среде *MS Excel* связано с примитивными приемами, когда в ячейки записываются формулы с необозримой длиной выражения со значениями соседних ячеек, где нередко фигурируют условные логические операторы, которые часто являются источником ошибок. В данной публикации автор хотел поделиться опытом учебной работы по математическому моделированию в среде *MS Excel* на примере изучения движения планет. Главная цель, это демонстрация методов численного решения систем дифференциальных уравнений второго порядка в плане понимания идеи способа интегрирования с последующей реализацией в виде двух пользовательских *VBA* функций.

Рассмотрим наиболее простой вариант записи уравнений Ньютона с приведенными единицами измерения: $\frac{d^2\vec{r}}{dt^2} = -G\frac{\vec{r}}{r^3}$ в векторном виде, который преобразуется к более удобной координатной форме:

$$\begin{cases} \dfrac{d^2x}{dt^2} = -G\dfrac{x}{r^3}, \\ \dfrac{d^2y}{dt^2} = -G\dfrac{y}{r^3} \end{cases}$$

. Фактически это система дифференциальных уравнений кроме двух неизвестных координат траектории движения планеты вокруг Солнца содержит ещё пару неизвестных компонент скоростей: $v_x = \frac{dx}{dt}, v_y = \frac{dy}{dt}$. Система будет содержать четыре дифференциальных уравнения:

$$\begin{cases} \dfrac{dv_x}{dt} = -G\dfrac{x}{r^3}, \dfrac{dx}{dt} = v_x, \\ \dfrac{dv_y}{dt} = -G\dfrac{y}{r^3}, \dfrac{dy}{dt} = v_y \end{cases}$$

. Каждое уравнение интегрируется на временном интервале $[a,b]$ по формуле Ньютона-Лейбница:

$$x(b) = x(a) + \int_a^b x'(t)dt$$

. Если интервал достаточно мал, то интеграл с достаточной точностью можно вычислить по формуле площади трапеции:

$$\int_a^b x'(t)dt = \frac{x'(a) + x'(b)}{2}(b-a)$$

.
Проблема заключается в том, что обычно неизвестно значение $x'(t)$ в точке $t=b$. Поэтому сначала интеграл

считают по формуле площади прямоугольника, а потом пересчитывают с новым значением производной с помощью правой части дифференциального уравнения. Этот прием получил название модифицированного метода Эйлера [3].

Приведенный теоретический экскурс понадобился для быстрого перехода к программному коду двух пользовательских функций без приведения блок-схемы, которая занимает много лишнего места.

Первая *VBA* функция *Уравнение_Ньютона* имеет аргументы в виде значений текущих координат траектории, а значения в виде массива из двух компонент ускорения движения:

```
Function Уравнение_Ньютона(ByVal x As Double, ByVal y As Double) As Variant
Dim r As Double, ускорение() As Double
   ReDim ускорение(2)
   g = 10 'константа для закона тяготения Ньютона в приведенных единицах
   r = Sqr(x * x + y * y)
   ускорение(1) = -g * x / (r ^ 3)
   ускорение(2) = -g * y / (r ^ 3)
   Уравнение_Ньютона = ускорение
End Function
```

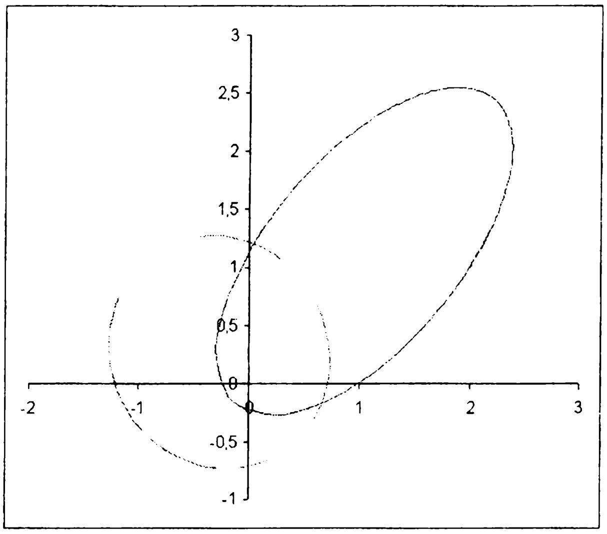
Обратим внимание на русские имена переменных, которые не допускаются ни в *ТурбоПаскале*, ни в *Delphi*. Константа для закона тяготения Ньютона $G=10$ с приведенными единицами измерения была выбрана чисто эмпирически из соображений соответствия начальных координат и скоростей величинам порядка единицы.

Вторая *VBA* функция: *Решение_уравнения_Ньютона* своими аргументами имеет n - число шагов интегрирования, h - шаг интегрирования, x_0, y_0 - начальные значения координат планеты и её v_{x0}, v_{y0} - компоненты скорости движения, соответственно. Значением функции является массив координат 100 точек траектории движения планеты, который используется затем на листе *MS Excel* для построения графика траектории.

```
Function Решение_уравнения_Ньютона(n As Long, h As Double, x0 As Double, y0 As Double, vx0 As Double, vy0 As Double) As Variant
Dim i, j, k As Long, ускорение() As Double, новое_ускорение() As Double, координаты(), x(), y(), vx(), vy() As Double
ReDim координаты_точек_траектории_для_графика(100, 2), x(n), y(n), скорость_x(n), скорость_y(n)
ReDim ускорение(2): ReDim новое_ускорение(2)
x(1) = x0: y(1) = y0
скорость_x(1) = vx0: скорость_y(1) = vy0
For i = 1 To n - 1
   ускорение = Уравнение_Ньютона(x(i), y(i))
   скорость_x(i + 1) = скорость_x(i) + ускорение(1) * h
   скорость_y(i + 1) = скорость_y(i) + ускорение(2) * h
   x(i + 1) = x(i) + h * (скорость_x(i) + скорость_x(i + 1)) / 2
   y(i + 1) = y(i) + h * (скорость_y(i) + скорость_y(i + 1)) / 2
   новое_ускорение = Уравнение_Ньютона(x(i + 1), y(i + 1))
   скорость_x(i + 1) = скорость_x(i) + (ускорение(1) + новое_ускорение(1)) * h / 2
   скорость_y(i + 1) = скорость_y(i) + (ускорение(2) + новое_ускорение(2)) * h / 2
   x(i + 1) = x(i) + h * (скорость_x(i) + скорость_x(i + 1)) / 2
   y(i + 1) = y(i) + h * (скорость_y(i) + скорость_y(i + 1)) / 2
Next i
k = n \ 100 'коэффициент для выборки 100 точек для графика из массива координат
For j = 1 To 100
   координаты_точек_траектории_для_графика(j, 1) = x(k * (j - 1) + 1)
   координаты_точек_траектории_для_графика(j, 2) = y(k * (j - 1) + 1)
Next j
Решение_уравнения_Ньютона = координаты_точек_траектории_для_графика
End Function
```

Обратим внимание на фрагмент кода функции: *новое_ускорение = Уравнение_Ньютона(x(i + 1), y(i + 1))*, где пересчитывается ускорение в новой точке траектории. Отметим также простоту описания атрибута значения функции: *As Variant*, за которым скрывается двухмерный большой массив длиной в несколько тысяч элементов, число которых задается первым параметром *n*.

Обе функции набираются в *VBA* редакторе, который запускается через кнопку *Сервис→Макрос→Редактор Visual Basic*. После набора вышеприведенного кода они становятся доступным пользователю через кнопку f_x. Для запуска достаточно выделить две колонки высотой в 100 ячеек в любом месте рабочего листа, далее через кнопку f_x задать функцию *Решение_уравнения_Ньютона* со следующими аргументами *Решение_уравнения_Ньютона (5000;0,001;2;1;1;1,5)*, смысл которых был выше разъяснен, после чего нажать комбинацию *<Shift>*+*<Ctrl>*+*<Enter>*, в результате выделенный массив будет заполнен значениями координат точек траектории движения планеты. С помощью этого массива через *Мастер диаграмм* выводим график вытянутой эллиптической траектории (см. рисунок). Для данной публикации на одном рабочем листе помещен ещё один массив координат почти круговой траектории движения другой планеты, соответствующий заданной функции *Решение_уравнения_Ньютона* со следующими параметрами аргументов *Решение_уравнения_Ньютона(3000;0,001;-1;1;-1,5;-1,5)*.



Заметим, что функция *Уравнение_Ньютона* нами явно не вызывалась, так как использовалась внутри функции *Решение_уравнения_Ньютона*. Эта гибкость была обусловлена возможностью перенастройки для решения других задач с одинаковой математической моделью в виде системы двух дифференциальных уравнений второго порядка. В этом случае достаточно переделать функцию, задающую правую часть уравнения Ньютона. Функция решения системы двух дифференциальных уравнений остается без изменений.

Литература

1. Поршнев С.В. Компьютерное моделирование физических процессов с использованием пакета MathCad.-М.:Горячая линия-Телеком, 2002.
2. Шнайдер Г. Справочник VBA 6.3.-М.:Лаборатория Базовых Знаний, 2002.
3. Гулд Х., Тобочник Я. Компьютерное моделирование в физике. Часть 1.-М.: Мир, 1990.

Локтев В.И., Михайлова М.А.

ИСПОЛЬЗОВАНИЕ МАТЕМАТИЧЕСКИХ ПРОГРАММ ПРИ ИЗУЧЕНИИ КУРСА ТЕОРЕТИЧЕСКОЙ МЕХАНИКИ

vilokt@rambler.ru
Астраханский государственный технический университет (АГТУ)
г. Астрахань

1. Выбор программы

В последнее время важной задачей учебных заведений является внедрение и широкое использование современных компьютерных программ. При реализации этой задачи у студентов не должно возникать представление, что каждый предмет существует только для самого себя (математика для математики, а современные инженерные и математические программы – для зачета по информатике). Изучение программ обоснованно, если студенты пользуются ими не только на уроках информатики, но и для решения задач других дисциплин. В ВУЗах к моменту изучения дисциплины «Теоретическая механика» студенты обладают необходимыми навыками пользователя оболочек *Windows 98/2000-XP* и программ *Microsoft Word, Mathcad, Excel, AutoCAD*. К сожалению, большинство современных инженерных программ (*Ansis, MathLab*), с помощью которых можно решать многие задачи теоретической и прикладной механики, не знакомы студентам второго курса. Выбор математических программ, в которых не заложены основы интересующей нас дисциплины, но позволяющие разрешить чисто математическую сторону задач, также достаточно широк. Самые распространенные *Maple, Mathematica, Mathcad*, но из них студенты знакомы, как правило, только с последней. Таким образом, в условиях сокращения часов на изучение общеинженерных дисциплин, при изучении дисциплины «Теоретическая механика» мы можем лишь упомянуть о существующих современных программах, их достоинствах и недостатках, ширине их возможностей и степени универсальности, но опереться сможем только на знакомый студентам *Mathcad*. Вопрос об обязательном выполнении работ на компьютере пока является спорным, но давать примеры выполнения заданий в *Mathcade* (на практических занятиях, в виде изданных методических указаний или вывешивать на стендах) и организовать систему поощрений для студентов, пользующихся компьютерными технологиями, безусловно нужно.