

Н.В.Емельянов

ПРАКТИЧЕСКАЯ НЕБЕСНАЯ МЕХАНИКА

Оглавление.

Глава 4. Построение моделей движения небесных тел. Методы численного интегрирования уравнений движения.

4.11. Численное интегрирование уравнений движения небесных тел.

4.11.1. Цели решения уравнений движения небесных тел

Методы численного интегрирования уравнений движения небесных тел разрабатываются и применяются для решения различных задач небесной механики. Совместно с аналитическими и качественными методами они являются процедурами, служащими целям практического познания природы.

Процесс изучения небесных тел состоит в построении модели их движения. Модель является ядром всех научных изысканий. Она постоянно уточняется на основе все новых наблюдений.

Что же представляет собой модель движения небесных тел? К настоящему времени мы уже знаем достаточно много о телах Солнечной системы. Открыты законы, согласно которым взаимодействуют планеты и спутники. Эти законы выражаются в форме дифференциальных уравнений относительно координат центров масс тел и относительно углов их вращения.

Использование модели заключается в предвычислении координат и угловых положений тел на любой заданный момент времени. Таким моментом может быть либо момент очередного наблюдения небесного тела, либо момент встречи космического аппарата с небесным телом, либо момент проведения наблюдений с искусственного космического объекта.

Предвычисление орбитального и вращательного движений небесных тел может делаться на основе аналитического решения дифференциальных уравнений движения, если такое решение имеется. Однако почти все практически значимые модели движения описываются уравне-

ниями, точное решение которых неизвестно. Что касается приближенных аналитических решений, то очень часто погрешность приближенного решения оказывается недопустимой, либо при требуемой точности процедура построения решения оказывается чрезмерно трудоемкой даже для мощнейших современных компьютеров.

Предвычисление координат и углов вращения тела на заданный момент времени может выполняться методами численного интегрирования обыкновенных дифференциальных уравнений. Преимущества численных методов по сравнению с аналитическими заключаются в относительной простоте их реализации в виде алгоритмов и вычислительных программ для компьютеров. Во многих случаях численные методы обладают большей точностью предвычисления координат небесных тел. Указанные преимущества обычно достигаются при высокой требуемой точности вычислений. Недостатками методов численного решения уравнений движения являются большие затраты времени вычислений, быстрый рост погрешности решения с ростом интервала предвычисления движения и невозможность достоверной оценки точности решения. При заданных уравнениях движения, начальных значениях координат тел и заданном моменте предвычисления точность решения оказывается ограниченной, сколь совершенным бы ни был применяемый численный метод решения.

Численные методы незаменимы в задачах, для которых невозможно получить аналитическое решение. Методы численного интегрирования служат также для проверки правильности и оценки точности найденного аналитического решения.

4.11.2. Общие свойства методов численного интегрирования уравнений движения

Постановка задачи о решении уравнений движения методами численного интегрирования распадается на два независимых этапа. Во-первых, составляются уравнения движения. Во-вторых, разрабатывается новый или выбирается один из известных методов численного интегрирования. Чтобы связать эти два этапа, необходимо принять некоторый стандартный вид уравнений движения.

Итак, в задачах численного интегрирования уравнений движения

небесных тел принята следующая форма уравнений:

$$\frac{dx_i}{dt} = f_i(x_1, x_2, \dots, x_n, t) \quad (i = 1, 2, 3, \dots, n), \quad (1)$$

где $x_i(t)$ являются искомыми функциями. Обязательным условием является задание функций $f_i(x_1, x_2, \dots, x_n, t)$ таким образом, чтобы их можно было вычислять при любых заданных значениях аргументов. Эти функции могут содержать также постоянные параметры, значения которых заданы.

К настоящему времени разработано огромное множество методов численного интегрирования обыкновенных дифференциальных уравнений. Они различны по своей эффективности и области приложений. Хороший обзор методов численного интегрирования уравнений движения, применяемых в небесной механике, дан в работе [1]. Существует даже теория методов численного решения уравнений, обобщающая множество таких методов и дающая путь разработки наиболее оптимальных схем интегрирования [2 – 5].

Разнообразные методы численного интегрирования при их применении обладают некоторыми общими свойствами. Эти свойства необходимо знать при решении конкретных задач для достижения наилучшего результата и эффективности исследований.

Основную идею методов численного интегрирования обыкновенных дифференциальных уравнений легко объяснить на простейшем из них – методе ломаных. При этом можно ограничиться рассмотрением одного дифференциального уравнения общего вида.

Допустим, что мы имеет одно дифференциальное уравнение

$$\frac{dx}{dt} = f(x, t),$$

в котором t есть независимая переменная – время. Требуется найти функцию $x(t)$, удовлетворяющую этому дифференциальному уравнению и начальному условию:

$$x(t_0) = x_0,$$

где t_0 – начальный момент времени, а x_0 – заданна константа.

Для начала важно понять то, что никакой метод численного интегрирования не позволит найти саму функцию $x(t)$, а только ряд ее значений на конечное число моментов времени. Рассмотрим некоторый

момент времени t_1 , недалеко отстоящий от момента t_0 . Разность $t_1 - t_0$ обозначим через h , то есть

$$h = t_1 - t_0.$$

Точное значение $x(t_1)$ нам неизвестно, но если величина h достаточно мала, а функция $f(x, t)$ - непрерывна, то приближенное значение x в момент времени t_1 можно определить по формуле

$$x_1 = x_0 + f(x_0, t_0) h.$$

Очевидно, что отличие x_1 от точного значения $x(t_1)$ будет тем меньше, чем меньше h . Погрешность обусловлена возможной нелинейностью функции $x(t)$ на отрезке $[t_0, t_1]$.

Далее мы можем определить приближенное значение функции $x(t)$ на момент времени $t_2 = t_1 + h$ по формуле

$$x_2 = x_1 + f(x_1, t_1) h.$$

Погрешность найденного значения искомой функции в момент t_2 будет складываться из погрешности x_1 и погрешности, вызванной нелинейностью функции $x(t)$ на отрезке $[t_1, t_2]$.

Теперь алгоритм определения значений искомой функции на последовательные моменты $t_i = t_{i-1}$ ($i = 1, 2, \dots$) можно записать так:

$$x_i = x_{i-1} + f(x_{i-1}, t_{i-1}) h. \quad (2)$$

В этом алгоритме величину h называют постоянным шагом численного интегрирования. Ошибка значения x_i будет содержать сумму ошибок, совершенных на всех предыдущих шагах интегрирования. Эти ошибки могут иметь различные величины и знаки, но при достаточно произвольной функции $f(x, t)$ суммарная ошибка, как правило, возрастает. Очевидно, что возрастание суммарной ошибки происходит с нарастанием числа выполненных шагов интегрирования. Чем больше шагов при постоянной величине шага, тем больше ошибка такого решения.

Любая задача численного интегрирования уравнений движения небесного тела обычно ставится так, что в конечном итоге требуется найти координаты тела на заданный конечный момент времени t_k . Если шаг интегрирования выбран, то можно подсчитать количество шагов k , необходимое для решения задачи. Очевидно, что

$$k = E \left(\frac{t_k - t_0}{h} \right) + 1.$$

Метод ломаных относится к группе так называемых одношаговых методов численного интегрирования. Кроме них существуют еще экспрополяционные и многошаговые методы [1]. Все эти методы имеют некоторые общие свойства. При их применении возникают общие проблемы. Одна из проблем – это выбор шага интегрирования. На первый взгляд ясно, что если шаг интегрирования уменьшить, то ошибка, допускаемая на каждом шаге, уменьшится, а следовательно повысится точность решения уравнений движения небесного тела. При заданном интервале времени предвычисления движения, то есть интервала интегрирования $t_k - t_0$, количество выполняемых шагов возрастет, что вызовет увеличение затрат вычислительного времени, то есть времени работы компьютера. В некоторых случаях максимальная точность решения определяется только допустимыми затратами времени вычислений.

На первый взгляд кажется, что если увеличение вычислительных затрат еще допустимо, то улучшения точности всегда можно достигнуть уменьшением шага интегрирования. Можно провести такой эксперимент. Задать конкретные уравнения движения, численное решение которых можно проверить. Это могут быть уравнения, имеющие точное аналитическое решение, или уравнения, допускающие известное частное решение. Далее можно задать интервал интегрирования $[t_0, t_k]$, взять какой-нибудь метод численного интегрирования и решать задачу многократно, уменьшая каждый раз шаг интегрирования. Изучая зависимость точности решения от шага интегрирования мы увидим, что сначала при уменьшении шага погрешность численного решения уменьшается. Но при некоторых весьма малых значениях шага интегрирования точность начнет ухудшаться, сколь бы мы ни уменьшали шаг. Что же происходит ?

Дело в том, что все вычисления делаются всегда с фиксированной точностью выполнения простых арифметических операций. Ее можно улучшить с помощью специальных способов отображения чисел в памяти компьютера и применением соответствующих алгоритмов арифметических действий. Это приведет к еще большим вычислительным затратам. Важно то, что в любом компьютере с соответствующим программным обеспечением максимальная точность представления чисел и точность выполнения арифметических фиксирована. При каждой операции совершается некоторая ошибка. Ее называют ошибкой округления чисел. При последовательном выполнении арифметических действий ошибки округления накапливаются.

Чем меньше шаг интегрирования, тем меньше погрешность формулы (2), меньше ошибка, совершающаяся на каждом шаге, и меньше ее накопление на конечный момент времени. С другой стороны, число шагов растет, и ошибки округления накапливаются. Таким образом существует некоторый оптимальный шаг интегрирования, при котором погрешность вычисления значения искомой функции на конечный момент времени минимальна. Для выбранного метода интегрирования и заданной конкретной задачи лучшая точность недостижима. Путем совершенствования методов численного интегрирования можно добиться некоторого улучшения точности. Однако современные методы уже столь совершенны, что дальнейшее повышение точности почти не происходит. На практике вычисления с оптимальных шагом, как правило, не производят – слишком велики при этом вычислительные затраты.

Следующую проблему вызывает тот факт, что степень нелинейности искомой функции может быть различна на разных участках интервала $[t_0, t_k]$. При заданном постоянном шаге интегрирования суммарная ошибка создается, в основном, на тех участках, где нелинейность максимальна. Нет смысла выбирать такой же малый шаг на других участках и тратить напрасно время на вычисления с излишней точностью. Оптимально было бы выбирать на каждом участке свой шаг интегрирования в зависимости от степени нелинейности искомой функции. В совершенных методах численного интегрирования поведение решения анализируется на предыдущем шаге, чтобы экономно выбрать следующий шаг. Для этого, конечно, перед началом интегрирования задают некоторый параметр, контролирующий точность вычислений на каждом шаге. Этот параметр называют заданной точностью интегрирования, однако эта точность далеко не совпадает с точностью получения решения на конечный момент времени t_k . Эти две точности обычно пропорциональны при одной и той же формулировке задачи, то есть при заданных уравнениях движения, начальных условиях и конечном моменте времени. Для анализа поведения решения используют различные приемы и формулы. Один из них состоит в том, что каждый шаг делают дважды: сначала один шаг величиной h , затем два шага, величиной $h/2$. Разницу двух результатов сравнивают с заданной константой – точностью вычислений. Если разность оказывается больше, то шаг уменьшают вдвое. Если же эта разность в десять или более раз меньше, чем заданная точность, то шаг увеличивают. В наиболее совершенных алгоритмах численного интегрирования используют более сложные приемы. К сожалению, на практике не всегда автоматический

выбор шага может быть эффективен. В некоторых случаях все же используют постоянный шаг интегрирования.

Наибольшую проблему в практике численного интегрирования уравнений движения небесных тел представляет оценка точности получаемого решения. Оказывается, не существует строгих формул или условий, позволяющих выполнить такие оценки. На деле используют некоторые приемы, которые все же нельзя считать вполне надежными. Один из них – это интегрирование "вперед-назад". То есть после получения значения искомой функции на конечный момент времени это значение принимают за исходное и выполняют интегрирование с отрицательным шагом до начального момента. В конце сравнивают полученный результат с начальным условием. Полученную разность значений и считают точностью выполненного интегрирования. Иногда такие оценки получаются удовлетворительными, но чаще всего реальная точность оказывается хуже. Существуют и другие методы оценки точности численного интегрирования, но все они на самом деле ненадежны.

4.11.3. Метод Рунге-Кутта интегрирования обыкновенных дифференциальных уравнений

Метод ломаных, изложенный в предыдущем параграфе, почти не применяется на практике. Существуют более точные методы. Одним из них является метод Рунге-Кутта [1, 6]. Этот метод далеко не самый совершенный, однако в свое время он широко применялся в астрономических задачах. Формулы метода Рунге-Кутта достаточно просты [6]. Их можно легко запрограммировать на любом подходящем языке программирования. В задачах, где не требуется получать предельно высокую точность численного решения уравнений движения, метод Рунге-Кутта может быть эффективно применен.

Опишем этот метод и формулы, по которым выполняются вычисления. Рассмотрим задачу численного интегрирования системы обыкновенных дифференциальных уравнений первого порядка

$$\frac{dx_i}{dt} = f_i(x_1, x_2, \dots, x_n, t) \quad (i = 1, 2, 3, \dots, n) \quad (3)$$

с начальными условиями $x_1 = x_1^{(0)}$, $x_2 = x_2^{(0)}$, ..., $x_n = x_n^{(0)}$ при $t = t_0$. Переменные x_1, x_2, \dots, x_n для удобства будем называть координатами, а независимую переменную t – временем.

Формулы Рунге-Кутта, приводимые ниже, позволяют определить координаты на момент времени $t_{k+1} = t_k + h$, если они известны на момент времени t_k . Формулы составлены на основе метода интерполяции полиномами относительно шага интегрирования h , в которых пренебрегают членами некоторого порядка малости относительно величины шага. В данном случае сохраняются все члены до 4-го порядка включительно относительно шага. Формулы Рунге-Кутта имеют вид

$$\begin{aligned} x_i^{(k)} &= x_i(t_k), \\ p_i^{(k)} &= f_i \left(x_1^{(k)}, x_2^{(k)}, \dots, x_n^{(k)}, t_k \right), \\ q_i^{(k)} &= f_i \left(x_1^{(k)} + \frac{1}{2} h p_1^{(k)}, x_2^{(k)} + \frac{1}{2} h p_2^{(k)}, \dots, x_n^{(k)} + \frac{1}{2} h p_n^{(k)}, t_k + \frac{1}{2} h \right), \\ r_i^{(k)} &= f_i \left(x_1^{(k)} + \frac{1}{2} h q_1^{(k)}, x_2^{(k)} + \frac{1}{2} h q_2^{(k)}, \dots, x_n^{(k)} + \frac{1}{2} h q_n^{(k)}, t_k + \frac{1}{2} h \right), \\ s_i^{(k)} &= f_i \left(x_1^{(k)} + h r_1^{(k)}, x_2^{(k)} + h r_2^{(k)}, \dots, x_n^{(k)} + h r_n^{(k)}, t_k + h \right), \\ x_i^{k+1} &= x_i^{(k)} + \frac{1}{6} h \left(p_i^{(k)} + 2q_i^{(k)} + 2r_i^{(k)} + s_i^{(k)} \right) \\ &\quad (i = 1, 2, 3, \dots, n). \end{aligned}$$

Из приведенных формул легко видеть схему алгоритма одношаговых методов численного интегрирования обыкновенных дифференциальных уравнений вида (3). Алгоритм состоит из двух относительно независимых блоков. В первом блоке производятся вычисления по формулам Рунге-Кутта. Входными данными для этого блока являются значения искомых функций на момент времени t_k . Результат работы блока – значения искомых функций на момент времени t_{k+1} . Разумеется, исходными параметрами являются также сами моменты t_k и t_{k+1} , а также шаг интегрирования h . Реализация этого блока не зависит от конкретной задачи небесной механики. Второй блок – вычисления правых частей уравнений (3) при любых заданных аргументах x_1, x_2, \dots, x_n, t . Этот блок полностью определяется постановкой задачи о движении или вращении небесных тел и не зависит от метода интегрирования.

4.11.4. Алгоритм решения задач о движении небесного тела методами численного интегрирования

Рассмотрим следующий класс задач о движении системы небесных

тел. Пусть даны дифференциальные уравнения относительно координат тел. Координатами могут быть как прямоугольные координаты центров масс каждого тела, так и углы поворота каждого тела в некоторой заданной системе координат. Уравнения должны быть выражены в форме (3), а их правые части заданы так, что известны правила, по которым их можно вычислять для любых заданных значений искомых координат и, возможно, времени. Кроме того, должны быть заданы значения координат на некоторый начальный момент времени t_0 . Обозначим эти значения через $x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)}$. Требуется определить значения координат на некоторый другой также заданный момент времени t_1 . Практические цели решения задачи о движении системы тел диктуют необходимость последовательного вычисления координат также и на другие заданные моменты t_2, t_3, \dots, t_k . Чаще всего моменты выбираются равноотстоящими так, что

$$t_i = t_{i-1} + H \quad (i = 1, 2, \dots, k), \quad (4)$$

где H – заданный шаг по времени (но не шаг интегрирования). Таким образом, результатом решения задачи будут значения искомых функций на моменты t_1, t_2, \dots, t_n , то есть $x_1^{(i)}, x_2^{(i)}, \dots, x_n^{(i)} \quad (i = 1, 2, \dots, k)$. Поставленная задача является типичной не только в практической небесной механике, но также при различных исследованиях теоретического характера.

Соответственно сформулированной постановке задачи можно составить алгоритм ее решения. Он будет состоять из следующих блоков:

- (1) задание начального момента времени, начальных значений координат и шага следования моментов H ;
- (2) задание следующего момента времени;
- (3) выполнение численного интегрирования уравнений и получение значений искомых функций на следующий момент времени;
- (4) запоминание полученных значений в файле для последующего использования;
- (5) проверка достижения последнего заданного момента времени и переход к пункту (2), если последний момент еще не достигнут;
- (6) прекращение работы алгоритма.

Пункт (1.) выполняется путем ввода чисел из файла исходных данных или заданием значений прямо в тексте вычислительной программы.

Пункт (2.) выполняется по формуле (4).

Реализация пункта (3.) в вычислительной программе должна быть независима от поставленной задачи, чтобы любую конкретную задачу можно было решать любым методом численного интегрирования. Поэтому сам метод интегрирования оформляют в виде отдельной процедуры на выбранном языке программирования. Эту процедуру обычно составляют специалисты по методам численного интегрирования обычновенных дифференциальных уравнений. Для использования процедуры в конкретной задаче нужно знать только форму обращения к этой процедуре и способы обмена данными с ней. Необходимая информация обычно содержится в инструкции, которой снабжается процедура при ее передаче или публикации. Пункт (3.) алгоритма реализуется согласно инструкции к процедуре. Пример такой инструкции приводится в следующем параграфе.

На первый взгляд в описанном выше алгоритме совершенно отсутствует блок вычисления правых частей решаемых уравнений. На самом деле этот блок включен в пункт (3.), но реализуется он в виде независимой процедуры, которую составляет специалист, поставивший и решающий задачу. Эта процедура используется процедурой численного интегрирования, которая не зависит от вида правых частей уравнений. Чтобы организовать составление алгоритма и программы, устанавливаются некоторые правила, по которым происходит обращение к процедуре вычисления правых частей уравнений движения. Эти правила также являются частью инструкции к процедуре интегрирования. Алгоритм процедуры вычисления правых частей уравнений не зависит от метода интегрирования, однако заголовок этой процедуры должен быть согласован с инструкцией к процедуре численного интегрирования.

Пункты (4.), (5.) и (6.) алгоритма достаточно просты для программирования. Заметим только, что пункт (4.) может быть реализован более замысловатым способом. Например, можно запрограммировать построение графических изображений на экране компьютера в зависимости от полученных значений искомых функций. В итоге последовательная смена картинок в процессе решения задачи даст наглядное представление о движении системы тел.

Заметим также, что при использовании некоторых процедур численного интегрирования необходимо задать определенные параметры

работы процедуры. Это могут быть, например, параметры, задающие точность интегрирования. В то же время результатом выполнения процедуры могут быть некоторые вспомогательные данные, которые также можно использовать в программе решения задачи.

4.11.5. Инструкция к вычислительной программе численного интегрирования обыкновенных дифференциальных уравнений методом Э. Эверхарта

В практической небесной механике применяются много различных методов численного интегрирования уравнений движения. Она дает стимул для развития и совершенствования таких методов. В то же время задачи небесной механики являются своеобразным полигоном для испытания новых разработок в этой области. Одним из наиболее применяемых в последнее время в небесной механике методов численного интегрирования является метод Э. Эверхарта. Показательно, что его автором является специалист в области небесной механики. Метод подробно описан в книге [1], где даны также ссылки на оригинальные публикации Э. Эверхарта. Первоначально процедура интегрирования была составлена на языке программирования Фортран самим Эверхартом, поэтому ее так и называют процедурой Эверхарта.

Основным достоинством метода Эверхарта является высокая достижимая точность интегрирования. Платой за точность оказываются большие затраты времени вычислений. Процедура интегрирования по методу Эверхарта составлена так, что с помощью двух параметров может быть задана необходимая точность вычислений. В зависимости от заданной точности изменяются необходимые затраты времени вычислений. При низкой требуемой точности вычислительные затраты будут небольшими, однако наилучшее соотношение точности вычислений и затрат времени достигается именно при высокой заданной точности. Основной параметр ε , определяющий точность вычислений на каждом шаге, выражают в виде

$$\varepsilon = 10^l, \quad (5)$$

где l – некоторое заданное целое число.

В методе Эверхарта используются специальные аппроксимирующие полиномы по степеням шага интегрирования. Степень этих полиномов N_{order} может быть выбрана из следующего списка: 7, 11, 15, 19, 23, 27.

В процедуре можно взять любое из этих значений. Однако при невысокой заданной точности интегрирования бесполезно задавать высокую степень полиномов. Это может привести лишь к неоправданным затратам времени вычислений. Поэтому N_{order} выбирают в зависимости от заданной точности вычислений в пределах

$$\frac{3}{4}l \leq N_{order} \leq 2l, \quad (6)$$

где l – целое число, фигурирующее в формуле (5).

Процедура Эверхарта позволяет решать уравнения движения как в режиме автоматического выбора шага интегрирования, так и при постоянном заданном шаге. Поскольку в небесной механике большинство задач описываются обыкновенными дифференциальными уравнениями второго порядка, то для удобства программирования задачи процедура Эверхарта позволяет решать сразу уравнения второго порядка, не разлагая их на удвоенное число уравнений первого порядка вида (1).

Наибольшую часть времени интегрирования занимают вычисления правых частей уравнений. Поэтому эффективность процедуры характеризуется количеством обращений к вычислению правых частей уравнений в конкретной задаче. Для оценки эффективности процедуры оказывается интересным также количество выполненных шагов интегрирования при автоматическом выборе шага. Обе количественные характеристики выдаются процедурой через ее выходные параметры.

Метод Эверхарта содержит процесс итераций при построении полиномов по степеням шага интегрирования на первом шаге. Обычно оказывается достаточным выполнить две итерации, но в некоторых задачах увеличение числа итераций на этом этапе алгоритма может привести к небольшому повышению эффективности процедуры без существенного увеличения затрат времени вычислений. Поэтому входным параметром процедуры является указанное число итераций, с помощью которого можно управлять эффективностью в некоторых небольших пределах. На начальной стадии решения конкретной задачи мало что известно о свойствах уравнений с точки зрения применения процедуры Эверхарта. Поэтому без сомнений можно задать число итераций равным 2.

В зависимости от того, уравнения какого вида нужно решать методом Эверхарта, задают класс уравнений. Уравнения разделяются на три класса.

Уравнения первого класса имеют вид (1). Ко второму классу отно-

сятся уравнения второго порядка вида

$$\frac{d^2x_i}{dt^2} = f_i(x_1, x_2, \dots, x_n, \frac{dx_1}{dt}, \frac{dx_2}{dt}, \dots, \frac{dx_n}{dt}, t) \quad (i = 1, 2, 3, \dots, n). \quad (7)$$

Уравнения третьего класса отличаются от уравнений (7) только тем, что правые части не зависят от компонент скорости, то есть от первых производных по времени от искомых функций. Это наиболее встречающиеся в небесной механике уравнения. Класс таких уравнений по странной традиции обозначается как -2. Уравнения класса -2 имеют вид

$$\frac{d^2x_i}{dt^2} = f_i(x_1, x_2, \dots, x_n, t) \quad (i = 1, 2, 3, \dots, n). \quad (8)$$

После описания всех особенностей процедуры Эверхарта приведем здесь инструкцию для ее использования. Инструкция, конечно, зависит от используемого языка программирования. Обычно это один из языков процедурного типа, например, Фортран, Паскаль или Си. Предположим, что программа вычислений составляется на языке программирования Паскаль. Тогда обращение к процедуре в программе пользователя имеет вид

```
Rada27(X, V, TI, TF, XL, LL, NV,  
       NI, NF, NS, NCLASS, NOR);
```

Опишем тип и смысл каждого параметра. Удобно начать с входного параметра NV. Это параметр типа Integer. Он вызывается процедурой по значению, то есть при обращении к процедуре на месте этого параметра может стоять любое выражение типа Integer. Параметр NV задает число искомых функций n .

Параметр NCLASS задает класс решаемых уравнений. Он может иметь значения 1, 2 или -2 соответственно описанной выше классификации уравнений. Параметр NCLASS типа Integer. Он вызывается процедурой по значению.

Параметр X - одномерный массив, описанный в программе пользователя в виде

X: array [0.. <n>] of Double;

где $< n >$ - константа, определяющая число уравнений. Этот массив используется для хранения и передачи значений искомых функций. Перед обращением к процедуре в этом массиве должны содержаться начальные значения на момент t_0 . После обращения элементы массива X будут содержать значения искомых функций на конечный момент

t_k . Параметр X является массивом открытого типа, то есть его длина в программе пользователя может быть любой, однако он должен быть описан в указанном выше виде. При этом элемент массива X[0] в процедуре не используется.

Параметр V – одномерный массив, описанный в программе пользователя в виде

V: array [0..<n>] of Double;

где $< n >$ - константа, определяющая число уравнений. Этот массив используется только для уравнений классов 1 и -2 для хранения и передачи значений первых производных от искомых функций по времени. Перед обращением к процедуре в этом массиве должны содержаться начальные значения на момент t_0 . После обращения элементы массива V будут содержать значения производных от искомых функций (скоростей) на конечный момент t_k . Параметр V также, как и X, является массивом открытого типа, его длина в программе пользователя может быть любой, при этом элемент массива V[0] в процедуре не используется.

Параметры TI и TF типа Double задают начальный и конечный моменты времени t_0 и t_k , соответственно. Эти параметры передаются по значению.

Параметр LL типа Integer задает точность вычислений. Он соответствует величине l в формуле (5). Этот параметр передается по значению. Автоматический выбор шага интегрирования делается в том случае, если значение параметра LL задано положительным. Для включения режима постоянного шага интегрирования следует задать значение LL равным любому отрицательному числу. В этом режиме для задания величины постоянного шага интегрирования используется параметр XL типа Double. Этот входной параметр передается по значению. При автоматическом выборе шага интегрирования параметр XL не используется.

Для задания степени аппроксимирующих полиномов N_{order} служит входной параметр NOR типа Integer. Он передается в процедуру по значению. Для выбора значения этого параметра следует использовать неравенства (6).

Параметр NI типа Integer задает число итераций при определении аппроксимирующего полинома. Это входной параметр, он передается в процедуру по значению. Как уже указано выше, значение этого параметра может быть выбрано равным 2 , если наиболее оптимальное значение этого параметра неизвестно.

Выходные параметры NF и NS передают в программу пользовате-

ля после вычислений количество сделанных обращений к вычислению правых частей уравнений и количество выполненных шагов интегрирования, соответственно. Эти параметры передаются в процедуру по наименованию. На месте этих параметров могут стоять только идентификаторы переменных типа Integer.

Теперь рассмотрим то, как процедура численного интегрирования Эверхарта взаимодействует с процедурой вычисления правых частей уравнений, которую должен составлять пользователь. В соответствии с формой обращения к этой процедуре внутри процедуры Эверхарта ее заголовок в программе пользователя должен иметь вид

```
procedure Force(var Xc, Vc : array of Double; TM: Double;  
var Fc: array of Double );
```

Массивы Xc и Vc при входе в процедуру содержат значения аргументов функций, определяющих правые части решаемых уравнений. В массиве Xc хранятся значения координат, а в массиве Vc – значения производных от координат по времени. Аргументы располагаются в массивах согласно их номерам, при этом элементы массивов Xc[0] и Vc[0] никогда не используются. Кроме того, параметр Vc используется только тогда, когда решаются уравнения класса 2.

Входной параметр TM типа Double задает текущий момент времени – аргумент правых частей уравнений. Он передается в процедуру по значению. Если правые части не зависят от времени, то параметр TM не используется.

Результат вычислений правых частей уравнений пользователь должен поместить в элементах выходного массива Fc. Для этого последовательно используются элементы этого массива, начиная с первого. Нулевой элемент массива Fc не используется.

При составлении вычислительной программы численного решения дифференциальных уравнений движения небесных тел могут проявляться некоторые особенности связанные с применяемым языком программирования и другими средствами программирования, например, компилятором. В частности, необходимо знать, что процедура Эверхарта, записанная на языке Паскаль, использует для своей работы несколько рабочих массивов, которые должны быть описаны вне тела процедуры как глобальные переменные с фиксированными именами. Длина этих рабочих массивов зависит от количества искомых функций, то есть от значения n в формулах (1), (7) и (8). Проще всего зарезервировать некоторую длину этих массивов и помнить максимальное допустимое

число искомых функций, которое нельзя превышать. В версии процедуры, которую мы используем на языке Паскаль, упомянутые массивы объявляются в модуле, содержащем процедуру Rada27, но вне тела этой процедуры. Эти объявления имеют вид

```
var F1, FJ, Y, Z : array [0..60] of Double;  
    BE, BT, B : array [1..13,1..60] of Double;
```

В этом случае число 60, фигурирующее в объявлениях рабочих массивов, означает максимальное число искомых функций. Если необходимо интегрировать уравнения с большим числом искомых функций, то вместо числа 60 в объявлениях массивов нужно поставить большее число. При необходимости можно сэкономить занимаемую массивами память компьютера, если вместо 60 поставить реально используемое число искомых функций.

При программировании процедуры Force вычисления правых частей уравнений могут потребоваться параметры, которые вводятся или задаются в основной программе. Передачу значений этих параметров можно сделать только через глобальные переменные, объявленные в одном из модулей программы и связанные через интерфейсы модулей.

При использовании варианта процедуры Эверхарта на других языках программирования указанные выше рабочие массивы могут описываться иначе.

Литература

1. Бодовицьна Т.В. Современные численные методы в задачах небесной механики. – М.: Наука. 1984 Главная редакция физико-математической литературы, 136 с.
2. Butcher J.C. Coefficients for the Study of Runge-Kutta Integration Processes. – J. Austral. Math. Soc., 1963, v. 3, p. 185 - 201.
3. Butcher J.C. On the Runge-Kutta Processes of High Order. – J. Austral. Math. Soc., 1964, v. 4, p. 179 - 195.
4. Butcher J.C. Implicit Runge-Kutta Processes. – Math. Comp., 1964, v. 18, p. 50 - 64.
5. Современные численные методы решения обыкновенных дифференциальных уравнений / Под ред. Дж. Холла, Дж. Уатта. – М.: Мир, 1979, 312 с.
6. Справочное руководство по небесной механике и астродинамике / Под ред. Г.Н.Дубошина. – М.: Наука, 1976, 862 с.