

УДК 518.61

ДРОБНО-КВАДРАТИЧНАЯ АППРОКСИМАЦИЯ ПРИ ЧИСЛЕННОМ ДИФФЕРЕНЦИРОВАНИИ И ИНТЕГРИРОВАНИИ

В.К. Королев

Задача Коши для обыкновенных дифференциальных уравнений имеет вид:

$$\frac{dy}{dt} = f(t, y), \quad y(t_0) = y_0, \quad t_0 \leq t \leq T. \quad (I)$$

Здесь t — скаляр, y — скаляр или вектор; f — соответственно скалярная или векторная функция, удовлетворяющая стандартным условиям теорем существования и единственности.

Для численного решения задачи (I) существует целый набор методов как классических, так и современных. Однако практика показывает, что имеется настоятельная потребность в дальнейшем совершенствовании методов интегрирования. Так, при проектировании больших интегральных схем требуется интегрировать систему, содержащую несколько сотен уравнений, причем число таких интегрирований может быть тоже порядка нескольких сотен (например, при решении задач оптимизации — детерминированной и, особенно, статистической). В настоящее же время решаются задачи для систем с числом уравнений не больше двух-трех десятков.

1.0 сплайновых методах интегрирования. Классические методы численного решения задачи (I) не могут быть признаны вполне удовлетворительными по следующим соображениям. Одношаговые методы используют большое число вычислений правых частей, что снижает их эффективность. Многошаговые методы разработаны в основном для равномерных

сеток, что тоже не всегда эффективно. Классические методы интегрирования не дают приемлемого способа выбора оптимального шага интегрирования — в зависимости от поведения интегральной кривой — без потери точности и без "лишних" вычислений правых частей (как, например, в программах по методу Рунге-Кутты с автоматическим выбором шага).

Повышение эффективности алгоритмов численного интегрирования может быть достигнуто при помощи сплайнов [1]. Их применение дает следующие преимущества. Во-первых, явная зависимость решения от параметров метода и независимой переменной позволяет получать решение во всех промежуточных точках (это, быть может, не всегда важно, однако этого нет в классических методах — все они "сеточные"). Во-вторых, эта явная зависимость — на каждом шаге интегрирования — дает возможность формировать величину самого шага. В-третьих, сплайновый подход допускает использование для аппроксимации не только полиномиальной, но и — в некотором смысле — любой другой системы функций в качестве базисной.

Напомним кратко идею сплайнового интегрирования. Пусть в точке t_m известно y_m (а в силу уравнения (I) и $\dot{y}_m = \frac{dy}{dt}(t_m)$). На интервале $[t_m, t_{m+1}]$, где t_{m+1} задано или подлежит определению, решение ищется в виде

$$y(t) = S(t, \alpha_1, \dots, \alpha_n),$$

здесь S — базисная функция, а $\alpha_1, \dots, \alpha_n$ — неизвестные пока параметры.

Например, в простейшем случае квадратичного сплайна

$$S = y_m + \alpha_1(t - t_m) + \alpha_2(t - t_m)^2 \quad (2)$$

(парабола проходит через точку (t_m, y_m) с наклоном $\alpha_1 = \dot{y}_m$ в этой точке). Параметр α_2 определяется из условия удовлетворения сплайна S уравнению (I) в точке t_{m+1} :

$$\dot{S}(t_{m+1}) = f(t_{m+1}, S(t_{m+1})) \quad (2')$$

или

$$\dot{y}_m + 2\alpha_2(t_{m+1} - t_m) = f(t_{m+1}, y_m + \dot{y}_m(t_{m+1} - t_m) + \alpha_2(t_{m+1} - t_m)^2).$$

Это уравнение относительно α_2 может быть достаточно сложным, так что решать его приходится каким-либо численным методом.

В общем случае параметры $\alpha_1, \dots, \alpha_m$ сплайна могут быть определены из M условий, аналогичных (2') и написанных для точек интервала $[t_m, t_{m+1}]$, из которых одна — t_{m+1} — конечная. Получается система нелинейных уравнений, и основная трудность при построении алгоритмов — в численном решении этой системы.

2. Дробно-квадратичная аппроксимация. В качестве системы базисных функций S можно брать различные функции. Например, наряду с параболой (2) интересно рассмотреть гиперболу вида

$$S = \frac{y_m + \alpha_1(t - t_m)}{1 + \alpha_3(t - t_m)}. \quad (3)$$

Оказывается (см. пример ниже), что с помощью гиперболы (3) удается построить алгоритм решения задачи (I), работающий в некоторых случаях и быстрее и точнее аналогичного алгоритма, использующего параболу (2).

Рассмотрим дробно-квадратичную функцию

$$y = \frac{y_m + \alpha_1(t - t_m) + \alpha_2(t - t_m)^2}{1 + \alpha_3(t - t_m) + \alpha_4(t - t_m)^2}, \quad (4)$$

частными случаями которой являются параболы (2) и гипербола (3). Производную этой функции удобно представить в следующем виде:

$$\dot{y} = \frac{\alpha_1 + 2\alpha_2(t - t_m) - (\alpha_3 + 2\alpha_4(t - t_m))y}{1 + \alpha_3(t - t_m) + \alpha_4(t - t_m)^2}. \quad (5)$$

Мы будем предполагать, что в двух соседних точках сетки t_{m-1} и t_m решение задачи (I) уже найдено, т.е. известны y_{m-1} , $\dot{y}_{m-1}$, y_m и $\dot{y}_m$. Искомой величиной при пошаговом интегрировании является y_{m+1} . Кроме того, если речь идет о выборе шага $h_{m+1} = t_{m+1} - t_m$, то и значение t_{m+1} неизвестно.

Используя выражения (4) и (5), а также известные и неизвестные пока значения y и $\dot{y}$, выпишем следующую "суперсистему",

из которой можно получить все системы относительно $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ для всех возможных частных случаев:

$$\begin{cases} \alpha_1 - \alpha_2 h_m - \alpha_3 y_{m-1} + \alpha_4 (y_{m-1} h_m) = \frac{y_m - y_{m-1}}{h_m}, & (6.1) \end{cases}$$

$$\begin{cases} \alpha_1 - \alpha_2 (2h_m) - \alpha_3 (y_{m-1} + \dot{y}_{m-1} h_m) + \alpha_4 h_m (2y_{m-1} + \dot{y}_{m-1} h_m) = \dot{y}_{m-1}, & (6.2) \end{cases}$$

$$\begin{cases} \alpha_1 - \alpha_3 y_m = \dot{y}_m, & (6.3) \end{cases}$$

$$\begin{cases} \alpha_1 + \alpha_2 h_{m+1} - \alpha_3 y_{m+1} - \alpha_4 (y_{m+1} h_{m+1}) = \frac{y_{m+1} - y_m}{h_{m+1}}, & (6.4) \end{cases}$$

$$\begin{cases} \alpha_1 + \alpha_2 (2h_{m+1}) - \alpha_3 (y_{m+1} + \dot{y}_{m+1} h_{m+1}) - \alpha_4 h_{m+1} (2y_{m+1} + \dot{y}_{m+1} h_{m+1}) = \dot{y}_{m+1}. & (6.5) \end{cases}$$

Уравнения (6.1) и (6.4) выражают условия прохождения функции (4) соответственно через точки (t_{m-1}, y_{m-1}) и (t_{m+1}, y_{m+1}) , остальные уравнения — это условия равенства производных в точках t_{m-1} , t_m и t_{m+1} .

3. Интерполяция и численное дифференцирование. Рассмотрим задачу интерполяции кривой, проходящей через три точки: (t_{m-1}, y_{m-1}) , (t_m, y_m) и (t_{m+1}, y_{m+1}) , на основе представления (4). Эта задача решается с помощью уравнений (6.1) и (6.4), которые позволяют определять любые два параметра из четырех. Например, в случае параболы (2) отличны от нуля только α_1 и α_2 , для гиперболы (3) — α_1 и α_3 .

Имея выражения той или иной пары коэффициентов, с помощью соотношений (6.2), (6.3) и (6.5), читаемых справа налево, можно получить формулы для производной интерполирующей функции в точках t_{m-1} , t_m и t_{m+1} . Например, в точке t_m параболическая интерполяция дает

$$\dot{y}_m = \alpha_1 = \frac{h_m \frac{y_{m+1} - y_m}{h_{m+1}} + h_{m+1} \frac{y_m - y_{m-1}}{h_m}}{h_m + h_{m+1}}. \quad (7)$$

(В случае равномерной сетки $h_m = h_{m+1} = h$, и мы получаем известную формулу $\dot{y}_m = \frac{y_{m+1} - y_{m-1}}{2h}$.) Аналогично при гиперболической интерполяции имеем:

$$\dot{y}_m = \frac{\frac{y_{m+1} - y_m}{h_{m+1}} \cdot \frac{y_m - y_{m-1}}{h_m} (h_m + h_{m+1})}{y_{m+1} - y_{m-1}}. \quad (8)$$

Из всех шести возможных случаев: $(\alpha_1, \alpha_2), (\alpha_1, \alpha_3), \dots, (\alpha_3, \alpha_4)$ лишь один, именно (α_2, α_4) , не представляет интереса из-за своей специфики ($\dot{y}_m = 0$ — из (6.3)). Во всех остальных случаях можно построить соответствующие интерполяционные формулы и формулы численного дифференцирования (и интегрирования — см. ниже).

Сравним между собой полученные формулы (7) и (8) на каком-нибудь тестовом примере. Пусть $y = e^t$, и при разных t_{m-1} , t_m и t_{m+1} сопоставим точное значение производной e^{t_m} с $\dot{y}_m$, рассчитанным по формулам (7) и (8).

Т а б л и ц а I

t	e^t	Парабола		Гипербола	
		$\dot{y}_m$	$E = e^{t_m} - \dot{y}_m$	$\dot{y}_m$	$E = e^{t_m} - \dot{y}_m$
$t_{m-1} = 1,0$	2,71828				
$t_m = 1,5$	4,48169	4,918146	-0,436456	4,304532	0,177158
$t_{m+1} = 2,5$	12,18249				
$t_{m-1} = 1,2$	3,32012				
$t_m = 1,5$	4,48169	4,600465	-0,118775	4,426525	0,055165
$t_{m+1} = 2,0$	7,38906				

Из таблицы I видно, что в данном примере гиперболическая интерполяция дает более точные значения производной $\dot{y}(t_m)$, чем параболическая. Геометрически это ясно: экспонента ближе к гиперболе, чем к параболе. Этот пример показывает, что имеет смысл

использовать неполиномиальную интерполяцию при конструировании численных алгоритмов.

4. Ф о р м у л ы ч и с л е н н о г о и н т е г р и р о в а н и я. При построении алгоритма численного решения задачи (I) мы будем исходить из сплайновой идеи о сопряжении интегральной кривой в узлах сетки до производных некоторого порядка (в нашем случае — до первого, так как требование большей гладкости в узлах может быть некорректным).

Уравнение (6.5) "суперсистемы" будет использоваться постоянно, а (6.4), напротив, — никогда. Остальные уравнения (или некоторые из них) употребляются в зависимости от используемой информации в двух предыдущих точках: t_{m-1} и t_m . В самом общем случае — это y_{m-1} , $\dot{y}_{m-1}$ и $\dot{y}_m$. Таким образом, для четырех параметров: $\alpha_1, \alpha_2, \alpha_3$ и α_4 мы имеем систему четырех уравнений.

Заметим, что все способы численного интегрирования, которые возможно получить на основе сплайнового подхода, относятся к нелинейным: $\dot{y}_{m+1}$ есть $f(t_{m+1}, y_{m+1})$, так что искомая величина y_{m+1} входит под знак функции f .

Рассмотрим некоторые частные случаи.

1. Парабола с использованием $\dot{y}_m$. Определяя α_1 и α_2 из уравнений (6.3) и (6.5) и подставляя их в (4), получаем

$$y_{m+1} = y_m + h_{m+1} \frac{\dot{y}_m + \dot{y}_{m+1}}{2} \quad (9)$$

— формулу так называемого модифицированного метода Эйлера.

2. Гипербола с использованием $\dot{y}_m$. После соответствующих выкладок получается аналог модифицированного метода Эйлера:

$$y_{m+1} = y_m + h_{m+1} \sqrt{\dot{y}_m \dot{y}_{m+1}}. \quad (10)$$

В отличие от (9), формула интегрирования получается нелинейной; интересно заметить, что в (10) вместо среднего арифметического производных $\dot{y}_m$ и $\dot{y}_{m+1}$ стоит их среднее геометрическое.

3. Парабола с использованием y_{m-1} . Уравнения (6.1) и (6.5) относительно α_1 и α_2 приводят к формуле

$$y_{m+1} = \frac{y_{m-1}(-h_m^2) + y_m(h_m + h_{m+1})^2 + \dot{y}_{m+1} h_m h_{m+1} (h_m + h_{m+1})}{h_m(h_m + 2h_{m+1})}, \quad (11)$$

известной под названием "метод Шихмана" [2] и широко применяемой в радио- и микроэлектронике.

Обратим внимание на то, что хотя модифицированный метод Эйлера и метод Шихмана основаны на, казалось бы, разных геометрических представлениях, однако они эквивалентны между собой с точки зрения используемой аппроксимации. Эта эквивалентность немедленно следует из (9) и (II), если вместо производной $\dot{y}_m$ взять ее выражение через ординаты - (7).

Обычно ординатные формулы в приближенных вычислениях предпочтительнее, поскольку погрешность при получении производных больше, чем при получении самих функций. Этим, кстати, и объясняется большая популярность метода Шихмана, чем модифицированного метода Эйлера.

4. Гипербола с использованием y_{m-1} . С помощью тех же уравнений (6.1) и (6.5) приходим к формуле, аналогичной методу Шихмана:

$$y_{m+1} = y_m + \frac{\frac{y_m - y_{m-1}}{h_m}}{\frac{y_{m+1} - y_{m-1}}{h_{m+1} + h_m}} \dot{y}_{m+1} h_{m+1}. \quad (12)$$

Получаемые уравнения относительно y_{m+1} можно брать в неявной форме. Выражение типа (12), разрешенное (возможно, не полностью) относительно искомой величины, более удобно, если мы собираемся для решения применить, например, метод простой итерации или метод Вегстейна.

Мы не будем выписывать все возможные формулы интегрирования, содержащиеся в "суперсистеме" (6), они получаются аналогично тому, как было показано выше.

5. Выбор шага интегрирования. Задача выбора шага возникает неизбежно при построении любого практического алгоритма численного интегрирования. Ясно, что мелкий шаг ведет к большим затратам машинного времени. При этом точность численного решения может и не повышаться. Большой шаг снижает точность. Желательно каждый раз шаг интегрирования выбирать возможно большим, но без потери точности решения.

Эта задача далеко не простая, и хотя имеются теоретические работы по этому вопросу (см., например, [3,4]), они, тем не менее, не содержат удовлетворительных практических рекомендаций.

Традиционный способ автоматического выбора шага, применяемый на практике, основан на использовании оценки погрешности метода с помощью остаточного члена в тейлоровском разложении решения. Однако эта оценка обычно является завышенной, поэтому не позволяет оптимально увеличивать шаг интегрирования. Кроме того, погрешность метода составляет лишь часть полной погрешности и не позволяет достаточно верно судить о последней.

Самый простой практический подход к решению задачи автоматического выбора шага предложен в [5], где рекомендуется выбирать его величину так, чтобы приращения переменных y не превышали некоторой (экспериментально найденной из предварительных расчетов) величины. Очевидно, можно продолжить эту идею "на порядок глубже", то есть выбирать шаг таким, чтобы приращения производных $\dot{y}$ были ограничены (или достаточно малы). Во-первых, такая модернизация даст эффект на крутых, но близких к линейным участках интегральных кривых. Во-вторых, мы исходим из того, что одну и ту же (или близкую) систему дифференциальных уравнений придется интегрировать многократно, так что некоторое число интегрирований можно потратить (например, аналогично описанному в [6]) на выбор соответствующих констант алгоритма. Кстати, и в приводимых выше теоретических работах типа [4] такое предварительное интегрирование предполагается.

Итак, пусть

$$\dot{y}_{m+1} = \dot{y}_m + \delta, \quad (13)$$

где δ выражается через $\alpha_1, \dots, y_m, \dots$ следующим образом (см. (5) при $t = t_m$ и $t = t_{m+1}$):

$$\delta = \frac{\alpha_3(y_m - y_{m+1}) + (2(\alpha_2 - \alpha_4 y_{m+1}) - \alpha_3(\alpha_1 - \alpha_3 y_m))h_{m+1} - \alpha_4(\alpha_1 - \alpha_3 y_m)h_{m+1}^2}{1 + \alpha_3 h_{m+1} + \alpha_4 h_{m+1}^2}.$$

Условие малости приращения δ :

$$|\delta| \leq \varepsilon \quad (14)$$

будем использовать в его предельном варианте - в виде равенства, что даст возможность увеличивать шаг h_{m+1} до возможного

допустимого предела. Получается квадратное уравнение относительно h_{m+1} , которое мы запишем в виде:

$$h_{m+1} = \frac{\varepsilon - \alpha_3(y_m - y_{m+1}) + \alpha_4(\varepsilon + \alpha_1 - \alpha_3 y_m) h_{m+1}^2}{2(\alpha_2 - \alpha_4 y_{m+1}) - \alpha_3(\varepsilon + \alpha_1 - \alpha_3 y_m)}.$$

Уравнение опять-таки неявное, во-первых, из-за наличия справа h_{m+1}^2 , во-вторых, из-за зависимости $\alpha_1, \dots, \alpha_4$ от h_{m+1} (см. формулы выше).

Для параболического случая это соотношение предельно упрощается: $h_{m+1} = \frac{\varepsilon}{2|\alpha_2|}$ и имеет простой геометрический смысл (шаг выбирается обратно пропорционально "кривизне" интегральной кривой в точке t_m).

В случае гиперболической аппроксимации

$$h_{m+1} = \frac{-\frac{\varepsilon}{\alpha_3} + (y_m - y_{m+1})}{\varepsilon + \alpha_1 - \alpha_3 y_m},$$

причем ε берется со знаком "+" или "-" так, чтобы было $h_{m+1} > 0$.

6. А л г о р и т м. Итак, мы имеем систему неявных уравнений относительно h_{m+1} и y_{m+1} следующего вида:

$$\begin{cases} h_{m+1} = \varphi(h_{m+1}, y_{m+1}), \\ y_{m+1} = \psi(h_{m+1}, y_{m+1}) \end{cases} \quad (I5)$$

(остальные аргументы для простоты не указаны).

Решение такой системы (второе уравнение, вообще говоря, векторное) представляет известные трудности. Классический подход к такой задаче - применение, например, метода Ньютона - требует вычисления и обращения матрицы частных производных, что приводит к довольно громоздким алгоритмам.

С другой стороны, за счет выбора достаточно малого ε в условии (I4) можно добиться, чтобы отображение, соответствующее системе (I5), было сжатым, и тогда даже простая итерация будет приводить к решению этой системы.

Поскольку мы не заинтересованы в уменьшении ε , то есть h_{m+1} , то желательно иметь способ решения системы (I5) более мощный, чем простая итерация, но и не такой громоздкий, как метод Ньютона.

Мы используем идею метода Вегстейна, разработанного, правда, для скалярного уравнения (см. [7]), именно: на $(k+1)$ -й итерации при фиксированном $y_{m+1}^{(k)}$ сделаем один шаг по методу Вегстейна, получим $h_{m+1}^{(k+1)}$. То же сделаем для второго уравнения при фиксированном $h_{m+1}^{(k+1)}$ и т.д. В случае плохой сходимости можно уменьшить ε .

Если решается задача (I) для векторного уравнения, то при решении системы (I5) надо учитывать, что условие (I4) должно удовлетворяться по всем компонентам y^i , $i = 1, \dots, n$, и тогда в качестве h_{m+1} выбирается наименьшее из h_{m+1}^i .

За начальное приближение для решения (I5) можно взять следующее. Перепишем (I3) в виде $\dot{y}_{m+1} \approx \dot{y}_m + \ddot{y}_m h_{m+1}$, где $\ddot{y}_m \approx \frac{\dot{y}_m - \dot{y}_{m-1}}{h_m}$. Отсюда с помощью (I4) имеем:

$$h_{m+1}^{(0)} = \frac{\varepsilon}{|\dot{y}_m - \dot{y}_{m-1}|} h_m.$$

Для получения $y_{m+1}^{(0)}$ используем уравнения (6.1) и (6.2). В рассмотренных выше - параболическом и гиперболическом - случаях находим соответственно:

$$y_{m+1}^{(0)} = -y_{m-1} z_1 (1 + z_2) + y_m z_2^2 - \dot{y}_{m-1} h_{m+1}^{(0)} z_2,$$

$$\text{где } z_1 = \frac{h_{m+1}^{(0)}}{h_m}, \quad z_2 = 1 + z_1,$$

и

$$y_{m+1}^{(0)} = \frac{y_m z_3 + z_1 (y_m z_3 + y_{m-1} z_4)}{z_3 + z_1 (z_3 + z_4)},$$

$$\text{где } z_3 = \dot{y}_{m-1} h_m, \quad z_4 = y_{m-1} - y_m.$$

Наконец, о разгоне. Для вычисления значений y и $\dot{y}$ в точках t_{m-1} и t_m можно применить какой-либо из одношаговых методов (например, метод Рунге-Кутты).

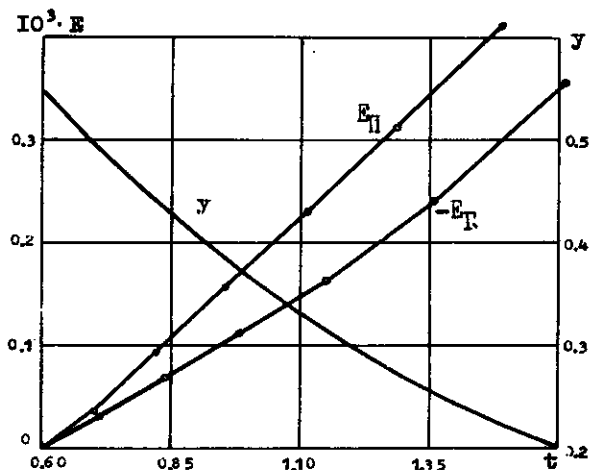
7. П р и м е р. Проиллюстрируем работу алгоритма на примере простой задачи: $\dot{y} = Ay$, $y(0) = y_0$, $0 \leq t \leq T$, точное решение которой есть $y = y_0 e^{At}$.

Исходные данные: $A = I$; $y_0 = 1$; $\varepsilon = 0,0625$; $T = 1,6$.

До $t = 0,6$

производится разгон по методу Рунге-Кутты.

Информация о решении с помощью параболической и гиперболической аппроксимации представлена в таблице 2. На рисунке показано поведение точного решения и функций ошибки для двух рассматриваемых аппроксимаций.



Из приведенных результатов видно, что по мере уменьшения кривизны решения шаг интегрирования увеличивается, причем несколько быстрее в гиперболическом случае. К тому же и ошибка в этом случае меньше, чем при параболической аппроксимации.

Итак, для повышения эффективности численных алгоритмов решения задачи Коши вместо распространенного приема увеличения порядка полиномиальной аппроксимации решения предлагается использовать неполиномиальную аппроксимацию, что приводит к некоторому набору алгоритмов, из которых - для каждой конкретной задачи - можно найти наилучший.

Вопросы теоретического обоснования получаемых таким путем нелинейных алгоритмов (сходимость, устойчивость, оценки) значительно усложняются и в данной работе не затрагивались. Прежде чем рассматривать эти вопросы, предполагается провести основательную проверку предлагаемого подхода при построении алгорит-

Таблица 2

П а р а б о л а					
№ пп	t	h_{n+1}	y	$\bar{y}$	$10^3 \cdot E_P$
0	0,6000000	-	0,5488168	-	-
1	0,7024099	0,1024099	0,4953900	0,4953507	0,0392571
2	0,8220449	0,1196349	0,4395319	0,4394369	0,0949372
3	0,9558144	0,1337695	0,3844988	0,3843403	0,1585278
4	1,107553	0,1517389	0,3303662	0,3301360	0,2302080
5	1,282495	0,1749428	0,2773441	0,2770303	0,3138165
6	1,488362	0,2058663	0,2257419	0,2253283	0,4136012
Г и п е р б о л а					
№ пп	t	h_{n+1}	y	$\bar{y}$	$-10^3 \cdot E_T$
0	0,6000000	-	0,5488168	-	-
1	0,7114250	0,1114250	0,4909440	0,4909766	0,0325944
2	0,8376253	0,1262003	0,4327359	0,4328080	0,0711698
3	0,9819759	0,1443506	0,3745702	0,3746839	0,1137293
4	1,150309	0,1683330	0,3165389	0,3167059	0,1670289
5	1,353195	0,2028866	0,2584131	0,2586535	0,2403855
6	1,610577	0,2573814	0,1997722	0,2001269	0,3546169

мов для решения реальных, достаточно серьезных задач, о которых говорилось ранее.

Л и т е р а т у р а

1. АЛБЕРТ Дж., НИЛЬСОН Э., УОЛШ Дж. Теория сплайнов и её приложения. М., "Мир", 1972.
2. SHICHMAN H. Integration system of a nonlinear network analysis program. - "IEEE Trans. on Circ.Th.", 1970, v. CT-17, N 3, p.378-385.
3. MORRISON D. Optimal mesh size in the numerical integration of an ordinary differential equation. - "J.ACM", 1962, v.9, N 1, p.98-103.
4. ТИХОНОВ А.Н., ГОРЕУНОВ А.Д., ГАЙСАРЯН С.С. Описание алгоритма построения оптимальных сеток при решении задачи Коши для обыкновенных дифференциальных уравнений методами Рунге-Кутты. - В кн.: Вычислительные методы и программирование, 1966, с. 17-20. (Моск. гос. ун-т.)

5. БЕЛОВ Б.И., НОРЕНКОВ И.П. Расчет электронных схем на ЭЦМ. М., "Машиностроение", 1971.

6. КОРОЛЕВ В.К. Сравнение некоторых методов численного интегрирования обыкновенных дифференциальных уравнений. — В кн.: Вычислительные системы. Вып. 64, Новосибирск, 1975, с. 108-127.

7. ЛАНС Д.Н. Численные методы для быстродействующих вычислительных машин. М., ИЛ, 1962.

Поступила в ред.-изд.отд.
6 мая 1975 года