ИЗВЕСТИЯ ВЫСШИХ УЧЕБНЫХ ЗАВЕДЕНИЙ
Т. 63, № 11 ФИЗИКА 2020

* *
*

УДК 519.62:531 DOI: 10.17223/00213411/63/11/131

В.А. АВДЮШЕВ

НОВЫЙ КОЛЛОКАЦИОННЫЙ ИНТЕГРАТОР ДЛЯ РЕШЕНИЯ ЗАДАЧ ДИНАМИКИ. I. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ [*]

Предлагается новый коллокационный интегратор на разбиении Лобатто для численного решения смешанных систем дифференциальных уравнений динамики первого и второго порядков. Излагается общая теория коллокационных интеграторов, из которой выводятся основные формулы нового интегратора.

Ключевые слова: *численные интеграторы, коллокационные методы, обыкновенные дифференциальные уравнения, динамические системы.*

Введение

Коллокационные интеграторы [1–4] представляют собой простой, изящный, удобный и вместе с тем мощный инструментарий для решения дифференциальных уравнений динамики. Идея коллокационных интеграторов весьма прозрачна и для ее понимания совершенно не требуется знаний других интеграторов [3], например, Рунге – Кутты, Грэгга – Булирша – Штера или Адамса. Достаточно лишь иметь представление, что такое дифференциальные уравнения, интегрирование и интерполирование. Поэтому вполне обоснованно рассматривать коллокационные интеграторы в целом как самостоятельный класс, из которого наиболее известными представителями являются неявные коллокационные интеграторы Рунге – Кутты.

Примечательной особенностью коллокационных интеграторов является то, что их теоретическая основа, как и программная реализация, универсальна для любого порядка [5]. Практически порядок определяется разбиением на шаге, а именно количеством и спецификой распределения узловых значений, через которые выражаются все остальные константы интегратора. Кроме того, на разбиениях гауссовых квадратур Лежандра и Лобатто коллокационные интеграторы становятся геометрическими [4]: симметричными и орбитально устойчивыми[**], а на разбиении Лежандра еще и симплектическими. Следует также отметить, что, в отличие от других интеграторов, коллокационные позволяют на каждом шаге легко конструировать приближенное аналитическое решение, чем удобно пользоваться для частого вывода результатов на плотной временной сетке.

В настоящей работе предлагается новый коллокационный интегратор Lobbie на разбиении Лобатто. Фактически его прототипом стал широко используемый в динамической астрономии интегратор Эверхарта [6, 7]. Точнее говоря, он является результатом кардинальной редакции своего предшественника, хотя в итоге теория, алгоритмизация и программный код нового интегратора могут лишь только концептуально напоминать об авторской версии прославленного интегратора Эверхарта.

Кратко излагается общая теория коллокационных интеграторов применительно к решению дифференциальных уравнений первого и второго порядков. Приводятся частные примеры, в том числе интегратор Эверхарта. Далее выводятся основные формулы интегратора Lobbie, а также отмечаются особенности в их программной реализации. Описывается программная процедура Lobbie.

1. Коллокационные методы

1.1. Дифференциальные уравнения первого порядка

Пусть динамическое состояние $\boldsymbol{x}$ описывается во времени t векторным дифференциальным уравнением первого порядка

[*] Работа выполнена при финансовой поддержке Минобрнауки РФ в рамках госзадания № 0721-2020-0049.

[**] Это определение применительно к методам Рунге – Кутты вводится в работе [5].

$$\boldsymbol{x}' = \boldsymbol{f}(t, \boldsymbol{x}) \tag{1}$$

при известном начальном динамическом состоянии на момент t_0:

$$\boldsymbol{x}_0 = \boldsymbol{x}(t_0). \tag{2}$$

Здесь штрих обозначает полную производную по времени; $\boldsymbol{f}$ – известная вектор-функция времени и динамического состояния. Необходимо определить динамическое состояние системы на момент $t_0 + h$:

$$\boldsymbol{x}(t_0 + h), \tag{3}$$

где h – малый параметр (длина шага интегрирования).

Представим приближенно решение уравнения (1) в виде полинома

$$\boldsymbol{u}(t) \approx \boldsymbol{x}(t), \tag{4}$$

который должен точно удовлетворять уравнению на некоторые промежуточные моменты времени $t_i \in [t_0, t_0 + h]$ $(i = 1, \ldots, s)$ (точки коллокаций)

$$\boldsymbol{u}'(t_i) = \boldsymbol{f}(t_i, \boldsymbol{u}(t_i)) \quad (i = 1, \ldots, s), \tag{5}$$

а также начальному условию (2)

$$\boldsymbol{x}_0 = \boldsymbol{u}(t_0). \tag{6}$$

Тогда решение (3) в соответствии с (4) приближенно определяется как

$$\boldsymbol{x}_1 = \boldsymbol{u}(t_0 + h) \approx \boldsymbol{x}(t_0 + h). \tag{7}$$

Геометрический смысл соотношений (5) состоит в том, что касательные к полиному в точках коллокаций должны совпадать (коллоцировать) с направлениями векторного поля, создаваемого функцией дифференциального уравнения $\boldsymbol{f}$ (рис. 1). При этом значения самого полинома могут заметно отличаться от точного решения.

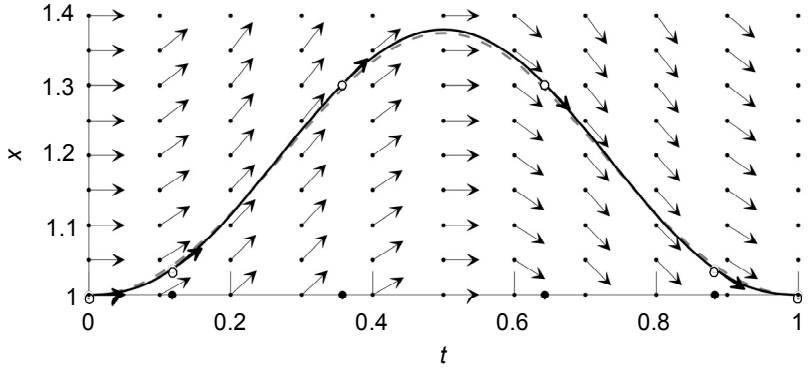

Рис. 1. Условия коллокаций для дифференциального уравнения $x' = f(t,x) = \sin(2\pi t)x$ при начальном условии $x_0 = x(0) = 1$. Стрелками указаны направления векторного поля $(\cos\alpha, \sin\alpha)$ в точках (t,x), где $\operatorname{tg}\alpha = f(t,x)$. Черная кривая – коллокационный полином u (приближенное решение) на разбиении Лобатто (черные точки) при $s = 6$; пунктирная кривая – точное решение $x = e^{(1-\cos(2\pi t))/2\pi}$; точки коллокаций на плоскости (t,x) выделены белым цветом

Условия коллокаций (5) можно рассматривать как условия Лагранжа, налагаемые на производную от полинома (4):

$$\boldsymbol{u}'(t_0 + h\tau) \equiv \boldsymbol{p}(\tau), \tag{8}$$

которая в таком случае выступает в роли полиномиального интерполянта функции $\boldsymbol{f}$ по безразмерной переменной τ. Согласно (5), условия Лагранжа на узловых значениях $c_1, \ldots, c_s$ безразмерной переменной τ можно представить в виде

$$\boldsymbol{p}(c_i) = \boldsymbol{f}_i \equiv \boldsymbol{f}(t_i, \boldsymbol{u}_i), \quad \boldsymbol{u}_i \equiv \boldsymbol{u}(t_i), \quad t_i = t_0 + hc_i \quad (i = 1, \ldots, s). \tag{9}$$

Формируя из условий (9) полиномиальный интерполянт $\boldsymbol{p}$ и затем интегрируя соотношение (8) по τ на отрезке $[0,1]$, учитывая (6) и (7):

$$\int_0^1 \boldsymbol{u}'(t_0 + h\tau)\,\mathrm{d}\,\tau = \frac{\boldsymbol{u}(t_0 + h\tau)\big|_0^1}{h} = \frac{\boldsymbol{x}_1 - \boldsymbol{x}_0}{h} = \int_0^1 \boldsymbol{p}(\tau)\,\mathrm{d}\,\tau;$$

получаем приближенное решение

$$\boldsymbol{x}_1 = \boldsymbol{x}_0 + h\int_0^1 \boldsymbol{p}(\tau)\,\mathrm{d}\,\tau.$$

Интерполянт $\boldsymbol{p}$ (8) конструируется из промежуточных приближенных решений $\boldsymbol{u}_1,\ldots,\boldsymbol{u}_s$ (9), т.е. $\boldsymbol{p} = \boldsymbol{p}(\tau,\boldsymbol{u}_1,\ldots,\boldsymbol{u}_s)$. Каждое i-е из этих решений также определяется путем интегрирования соотношения (8) по τ, но на отрезке $[0,c_i]$. Таким образом, коллокационный метод для решения дифференциального уравнения (1) можно представить как сводку формул

$$\boldsymbol{x}_1 = \boldsymbol{x}_0 + h\int_0^1 \boldsymbol{p}(\tau,\boldsymbol{u}_1,\ldots,\boldsymbol{u}_s)\,\mathrm{d}\,\tau, \quad \boldsymbol{u}_i = \boldsymbol{x}_0 + h\int_0^{c_i} \boldsymbol{p}(\tau,\boldsymbol{u}_1,\ldots,\boldsymbol{u}_s)\,\mathrm{d}\,\tau \quad (i = 1,\ldots,s). \tag{10}$$

Промежуточные решения в (10) задаются неявным образом через нелинейные уравнения. По этой причине все коллокационные методы – неявные*. Уравнения решаются, как правило, методом простых итераций в модификации Зейделя, т.е. поочередно уточняя промежуточные решения на каждой итерации. Фактически итерационное решение нелинейных уравнений является ядром любого коллокационного интегратора, и его эффективность во многом зависит от того, насколько удачно организован итерационный процесс.

Порядок метода p обусловлен количеством точек коллокаций s, а также спецификой их распределения. Принцип коллокаций позволяет получить практически любой порядок. Так, для произвольного распределения $c_1,\ldots,c_s$ (например, равномерного), по меньшей мере, $p = s$ [3]. Однако при использовании узловых значений гауссовых квадратур Лежандра, Радау и Лобатто порядок можно повысить до $p = 2s$, $p = 2s-1$ и $p = 2s-2$ соответственно [3, 4, 8, 9]. Эти узловые значения являются решениями алгебраических уравнений

$$\frac{\mathrm{d}^s}{\mathrm{d}\,\tau^s}[\tau^s(\tau-1)^s] = 0 \quad \text{(Legendre)};$$

$$\frac{\mathrm{d}^{s-1}}{\mathrm{d}\,\tau^{s-1}}[\tau^s(\tau-1)^{s-1}] = 0 \quad \text{(Radau I)}; \quad \frac{\mathrm{d}^{s-1}}{\mathrm{d}\,\tau^{s-1}}[\tau^{s-1}(\tau-1)^s] = 0 \quad \text{(Radau II)}; \tag{11}$$

$$\frac{\mathrm{d}^{s-2}}{\mathrm{d}\,\tau^{s-2}}[\tau^{s-1}(\tau-1)^{s-1}] = 0 \quad \text{(Lobatto)}.$$

Коллокационные методы примечательны тем, что фактически они дают аналитическое решение на шаге в форме коллокационного полинома

$$\boldsymbol{u}(\tau) = \boldsymbol{x}_0 + h\int_0^\tau \boldsymbol{p}(\tau)\,\mathrm{d}\,\tau$$

(а не только решение $\boldsymbol{x}_1$ на конце шага и несколько промежуточных $\boldsymbol{u}_1,\ldots,\boldsymbol{u}_s$), что очень удобно для вывода приближенных решений на плотной временной сетке. Впрочем, следует помнить, что порядок точности внутри шага понижается до $p = s$ [3]. Кроме того, наличие интерполянта позволяет путем экстраполирования получать достаточно хорошие начальные приближения функции $\boldsymbol{f}$ на следующем шаге:

$$\boldsymbol{f}_i = \boldsymbol{p}(1 + c_i) \quad (i = 1,\ldots,s)$$

для дальнейшего итерационного определения промежуточных решений (10).

Следует особо отметить, что на разбиениях Лежандра и Лобатто (11) коллокационные методы становятся геометрическими [4]: симметричными и орбитально устойчивыми, а на разбиении Лежандра еще и симплектическими. Казалось бы, разбиение Лежандра предпочтительней, поскольку при одном и том же количестве точек коллокаций его порядок на два выше. Однако, несмотря на то, что порядок метода на разбиении Лобатто ниже, он работает немного быстрее. Так, на каждом шаге при ni итерациях для определения промежуточных решений количество вычислений функции правых частей составляет $ncf = ni\cdot(s-1)$, тогда как на разбиении Лежандра – $ncf = ni\cdot s$.

* Кроме явного метода Эйлера (первого порядка), который является коллокационным на (левом) разбиении Радау при $s = 1$.

Кроме того, интерполянт правой части уравнения строится на всем отрезке интегрирования $[t_0, t_0+h]$, поскольку $c_1=0$ и $c_s=1$, в отличие от разбиения Лежандра, где все узловые значения находятся внутри отрезка. Следовательно, предиктор метода на разбиении Лобатто будет лучше, что очень важно для его программной реализации.

В качестве простых примеров коллокационных методов можно привести хорошо и давно известные методы Рунге – Кутты: явный и неявный Эйлера (Radau I & II: $s=1$, $p=1$); средней точки (Legendre: $s=1$, $p=2$); трапеции (Lobatto: $s=2$, $p=2$); Симпсона (Lobatto: $s=3$, $p=4$). Если в качестве интерполянта функции дифференциального уравнения принять полином Лагранжа

$$\boldsymbol{p}(\tau)=\sum_{j=1}^{s}\boldsymbol{f}_j\prod_{k\neq j}\frac{\tau-c_k}{c_j-c_k},$$

то коллокационный метод (10) принимает классическую форму метода Рунге – Кутты [1–4]:

$$\boldsymbol{x}_1=\boldsymbol{x}_0+h\sum_{j=1}^{s}b_j\boldsymbol{f}_j,\quad \boldsymbol{u}_i=\boldsymbol{x}_0+h\sum_{j=1}^{s}a_{ij}\boldsymbol{f}_j\quad (i=1,\ldots,s), \tag{12}$$

где постоянные выражаются через интегралы от базисных функций Лагранжа:

$$a_{ij}=\int_0^{c_i}\prod_{k\neq j}\frac{\tau-c_k}{c_j-c_k}\mathrm{d}\tau,\quad b_j=\int_0^{1}\prod_{k\neq j}\frac{\tau-c_k}{c_j-c_k}\mathrm{d}\tau.$$

Вскоре после первых работ по коллокационным методам Рунге – Кутты [1, 2] Э. Эверхарт [6, 7] предложил в качестве интерполянта использовать полином в каноническом виде*

$$\boldsymbol{p}(\tau)=\sum_{j=1}^{s}\boldsymbol{a}_j\tau^{j-1}.$$

Здесь $\boldsymbol{a}_j=\boldsymbol{a}_j(\boldsymbol{f}_1,\ldots,\boldsymbol{f}_s)$ $(j=1,\ldots,s)$. Простая форма интерполяции позволяет получить достаточно простую форму приближенного решения

$$\boldsymbol{x}_1=\boldsymbol{x}_0+h\sum_{j=1}^{s}\frac{\boldsymbol{a}_j}{j},\quad \boldsymbol{u}_i=\boldsymbol{x}_0+h\sum_{j=1}^{s}\frac{\boldsymbol{a}_j}{j}c_i^j\quad (i=1,\ldots,s). \tag{13}$$

Однако, чтобы выразить коэффициенты полинома $\boldsymbol{a}$ через коллокационные значения функции $\boldsymbol{f}$, автор прибегает к разделенным разностям интерполяционного полинома Ньютона $\boldsymbol{\alpha}$, которые непосредственно определяются из коллокационных значений. Между тем коэффициенты канонического полинома выражаются через разделенные разности посредством линейных соотношений

$$\boldsymbol{a}_j=\sum_{i=j}^{s}c_{ij}\boldsymbol{\alpha}_i\quad (j=1,\ldots,s),$$

где постоянные вычисляются из узловых значений разбиения $c_1,\ldots,c_s$ по рекуррентным формулам

$$c_{ii}=1\quad (i>0),\quad c_{ij}=c_{i-1,j-1}-c_{i-1}c_{i-1,j}\quad (i>j>0).$$

1.2. Дифференциальные уравнения второго порядка

Предположим теперь, что динамическая система описывается векторным дифференциальным уравнением второго порядка

$$\boldsymbol{x}''=\boldsymbol{f}(t,\boldsymbol{x},\boldsymbol{x}') \tag{14}$$

при известном начальном динамическом состоянии на момент t_0:

$$\boldsymbol{x}_0=\boldsymbol{x}(t_0),\quad \boldsymbol{x}'_0=\boldsymbol{x}'(t_0). \tag{15}$$

Необходимо определить динамическое состояние системы на момент t_0+h:

$$\boldsymbol{x}(t_0+h),\quad \boldsymbol{x}'(t_0+h). \tag{16}$$

Здесь $\boldsymbol{x}$ и $\boldsymbol{x}'$ будем рассматривать как векторы координат и скоростей соответственно.

Представим приближенно решения для координат и скоростей в виде полиномов

$$\boldsymbol{u}(t)\approx\boldsymbol{x}(t),\quad \boldsymbol{v}(t)=\boldsymbol{u}'(t)\approx\boldsymbol{x}'(t), \tag{17}$$

которые должны удовлетворять уравнению (14) на моменты коллокаций $t_i\in[t_0,t_0+h]$ $(i=1,\ldots,s)$:

* Хотя сам автор не представлял свой метод как коллокационный.

$$\boldsymbol{u}''(t_i) = \boldsymbol{v}'(t_i) = \boldsymbol{f}(t_i, \boldsymbol{u}(t_i), \boldsymbol{v}(t_i)) \quad (i = 1,\ldots,s), \tag{18}$$

а также начальному условию (15):

$$\boldsymbol{x}_0 = \boldsymbol{u}(t_0), \quad \boldsymbol{x}_0' = \boldsymbol{v}(t_0). \tag{19}$$

Тогда решения (16) в соответствии с (17) приближенно определяются как

$$\boldsymbol{x}_1 = \boldsymbol{u}(t_0 + h) \approx \boldsymbol{x}(t_0 + h), \quad \boldsymbol{x}_1' = \boldsymbol{v}(t_0 + h) \approx \boldsymbol{x}'(t_0 + h). \tag{20}$$

Согласно (18), условия Лагранжа для интерполянта $\boldsymbol{p}$ функции $\boldsymbol{f}$

$$\boldsymbol{u}''(t_0 + h\tau) = \boldsymbol{v}'(t_0 + h\tau) \equiv \boldsymbol{p}(\tau), \tag{21}$$

можно представить в виде

$$\boldsymbol{p}(c_i) = \boldsymbol{f}_i \equiv \boldsymbol{f}(t_i, \boldsymbol{u}_i, \boldsymbol{v}_i), \quad \boldsymbol{u}_i \equiv \boldsymbol{u}(t_i), \quad \boldsymbol{v}_i \equiv \boldsymbol{v}(t_i), \quad t_i = t_0 + hc_i \quad (i = 1,\ldots,s). \tag{22}$$

Формируя из условий (22) полиномиальный интерполянт $\boldsymbol{p}$ и затем интегрируя соотношение (21) по τ на отрезках $[0,1]$ и $[0,c_i]$ $(i = 1,\ldots,s)$, получаем сводку формул для коллокационного метода применительно к дифференциальному уравнению (14):

$$\begin{gathered} \boldsymbol{x}_1 = \boldsymbol{x}_0 + \boldsymbol{x}_0' h + h^2 \textstyle\int_0^1 \int_0^\tau \boldsymbol{p}(\tau)\,\mathrm{d}\tau^2, \qquad \boldsymbol{x}_1' = \boldsymbol{x}_0' + h \textstyle\int_0^1 \boldsymbol{p}(\tau)\,\mathrm{d}\tau, \\ \boldsymbol{u}_i = \boldsymbol{x}_0 + \boldsymbol{x}_0' h c_i + h^2 \textstyle\int_0^{c_i} \int_0^\tau \boldsymbol{p}(\tau)\,\mathrm{d}\tau^2, \quad \boldsymbol{v}_i = \boldsymbol{x}_0' + h \textstyle\int_0^{c_i} \boldsymbol{p}(\tau)\,\mathrm{d}\tau \quad (i = 1,\ldots,s), \end{gathered} \tag{23}$$

где подынтегральный полином $\boldsymbol{p}$ зависит также от промежуточных решений $\boldsymbol{u}_1,\ldots,\boldsymbol{u}_s$ и $\boldsymbol{v}_1,\ldots,\boldsymbol{v}_s$.

1.3. Смешанные системы дифференциальных уравнений первого и второго порядков

Задачу (14) и (15) можно представить в виде (1) и (2):

$$\boldsymbol{x}' = \dot{\boldsymbol{x}}, \quad \dot{\boldsymbol{x}}' = \boldsymbol{f}(t, \boldsymbol{x}, \dot{\boldsymbol{x}}), \quad \boldsymbol{x}_0 = \boldsymbol{x}(t_0), \quad \dot{\boldsymbol{x}}_0 = \dot{\boldsymbol{x}}(t_0) = \boldsymbol{x}_0'. \tag{24}$$

Несмотря на то, что обе задачи описывают одну и ту же динамическую систему, их численные решения в соответствии с принципом коллокаций будут принципиально разными. Так, согласно (10), для альтернативной задачи будем иметь приближенные решения

$$(\boldsymbol{x}_1, \dot{\boldsymbol{x}}_1) = (\boldsymbol{x}_0, \dot{\boldsymbol{x}}_0) + h \textstyle\int_0^1 (\boldsymbol{q}(\tau), \boldsymbol{p}(\tau))\,\mathrm{d}\tau, \quad (\boldsymbol{u}_i, \boldsymbol{v}_i) = (\boldsymbol{x}_0, \dot{\boldsymbol{x}}_0) + h \textstyle\int_0^{c_i} (\boldsymbol{q}(\tau), \boldsymbol{p}(\tau))\,\mathrm{d}\tau \quad (i = 1,\ldots,s), \tag{25}$$

где полином $\boldsymbol{q}$ интерполирует правую часть уравнения для вектора координат. Точность решений (25) (как для координат, так и для скоростей) – одного порядка p. Однако в (23) решение для координат вследствие двойного интегрирования имеет порядок $p+1$, иначе говоря, координаты в (23) определяются точнее, нежели в (25).

Тем не менее пользователь порой вынужден обращаться к представлению динамической системы в виде (24) и тем самым прибегать к интегратору для уравнений первого порядка (25), вполне осознавая, что это неизбежно влечет значительное понижение точности приближенного решения. Такая необходимость возникает, когда уравнение динамической системы (14) дополняется уравнениями первого порядка для каких-либо вспомогательных динамических величин. Например, смешанные системы применяются для исследования динамического хаоса [10], а также для линеаризации, регуляризации и стабилизации уравнений динамики [11–14]. Так, чтобы привести все уравнения смешанной системы к единому порядку, вместо (14) используются уравнения (24). Альтернативный вариант – это дифференцирование дополнительных уравнений, с тем чтобы вся система в целом имела вид (14). Хотя для сложных динамических систем взятие производных от функций дополнительных уравнений часто практически невозможно.

Естественный и эффективный подход к решению смешанной системы уравнений – это применение гибридного интегратора. Если для моделирования динамической системы необходимо совместно с (14) решать векторное уравнение первого порядка для вспомогательных динамических величин $\boldsymbol{z}$:

$$\boldsymbol{z}' = \boldsymbol{g}(t, \boldsymbol{x}, \boldsymbol{x}', \boldsymbol{z}), \tag{26}$$

при начальном условии $\boldsymbol{z}_0 = \boldsymbol{z}(t_0)$, сводку (23) следует дополнить формулами

$$\boldsymbol{z}_1 = \boldsymbol{z}_0 + h \textstyle\int_0^1 \boldsymbol{r}(\tau)\,\mathrm{d}\tau, \quad \boldsymbol{w}_i = \boldsymbol{z}_0 + h \textstyle\int_0^{c_i} \boldsymbol{r}(\tau)\,\mathrm{d}\tau \quad (i = 1,\ldots,s). \tag{27}$$

Здесь интерполянт $\boldsymbol{r}$ функции $\boldsymbol{g}$ конструируется по аналогии с интерполянтом $\boldsymbol{p}$ из промежуточных решений $\boldsymbol{u}_1,\ldots,\boldsymbol{u}_s,\boldsymbol{v}_1,\ldots,\boldsymbol{v}_s$ и $\boldsymbol{w}_1,\ldots,\boldsymbol{w}_s$. Таким образом, гибридный коллокационный метод для уравнений (14) и (26) будет иметь вид (23) и (27).

2. Интегратор Lobbie

Применительно к смешанной системе дифференциальных уравнений (14) и (26):

$$\boldsymbol{x}'' = \boldsymbol{f}(t,\boldsymbol{x},\boldsymbol{x}',z), \quad z' = \boldsymbol{g}(t,\boldsymbol{x},\boldsymbol{x}',z), \quad \boldsymbol{x}_0 = \boldsymbol{x}(t_0), \quad \boldsymbol{x}'_0 = \boldsymbol{x}'(t_0), \quad z_0 = z(t_0), \tag{28}$$

примем в качестве интерполянтов полиномы Ньютона

$$\boldsymbol{p}(\tau) = \sum_{j=1}^{s} \boldsymbol{\alpha}_j \prod_{k=1}^{j-1}(\tau - c_k) \quad \text{и} \quad \boldsymbol{r}(\tau) = \sum_{j=1}^{s} \boldsymbol{\beta}_j \prod_{k=1}^{j-1}(\tau - c_k) \tag{29}$$

на разбиении Лобатто $(c_1 = 0, c_s = 1)$. Здесь $\Pi_{k=1}^{0} = 1$. Разделенные разности в (29) определяются из узловых значений функций $\boldsymbol{f}$ и $\boldsymbol{g}$ по рекуррентным формулам

$$\begin{gathered} \boldsymbol{\alpha}_j = \boldsymbol{f}_j, \quad \boldsymbol{\beta}_j = \boldsymbol{g}_j, \quad \boldsymbol{\alpha}_j := (\boldsymbol{\alpha}_j - \boldsymbol{\alpha}_k)/(c_j - c_k), \quad \boldsymbol{\beta}_j := (\boldsymbol{\beta}_j - \boldsymbol{\beta}_k)/(c_j - c_k) \\ (j = 1,\ldots,s;\ k = 1,\ldots,j-1). \end{gathered} \tag{30}$$

Подставляя интерполянты (29) в (23) и (27), а также учитывая, что $c_1 = 0$ и $c_s = 1$, получим коллокационный метод для системы (28) в виде

$$\begin{gathered} \boldsymbol{u}_1 = \boldsymbol{x}_0, \quad \boldsymbol{v}_1 = \boldsymbol{x}'_0, \quad \boldsymbol{w}_1 = \boldsymbol{z}_0, \\ \boldsymbol{u}_i = \boldsymbol{x}_0 + \boldsymbol{x}'_0 h c_i + h^2 \sum_{j=1}^{s} a_{ij}\boldsymbol{\alpha}_j, \quad \boldsymbol{v}_i = \boldsymbol{x}'_0 + h\sum_{j=1}^{s} b_{ij}\boldsymbol{\alpha}_j, \quad \boldsymbol{w}_i = \boldsymbol{z}_0 + h\sum_{j=1}^{s} b_{ij}\boldsymbol{\beta}_j \quad (i = 2,\ldots,s), \\ \boldsymbol{x}_1 = \boldsymbol{u}_s, \quad \boldsymbol{x}'_1 = \boldsymbol{v}_s, \quad \boldsymbol{z}_1 = \boldsymbol{w}_s, \end{gathered} \tag{31}$$

где

$$a_{ij} = \int_0^{c_i}\int_0^{\tau}\prod_{k=1}^{j-1}(\tau - c_k)\,\mathrm{d}\tau^2, \quad b_{ij} = \int_0^{c_i}\prod_{k=1}^{j-1}(\tau - c_k)\,\mathrm{d}\tau \quad (i,j = 1,\ldots,s). \tag{32}$$

Обозначим k-кратный интеграл от j-й базисной функции Ньютона как

$$\gamma_{jk}(\tau) = \int_0^{\tau}\ldots\int_0^{\tau}(\tau - c_1)\ldots(\tau - c_{j-1})\,\mathrm{d}\tau^k\,. \tag{33}$$

Если величины $\gamma_{jk}(\tau)$ $(j,k = 1,\ldots,s+1)$ рассматривать как элементы некой матрицы Γ размера $(s+1)\times(s+1)$, то константы коллокационного метода – интегралы от базисных функций Ньютона (32) – будут составлять первые ее два столбца:

$$a_{ij} = \gamma_{j2}(c_i), \quad b_{ij} = \gamma_{j1}(c_i) \quad (i,j = 1,\ldots,s).$$

Между тем элементы матрицы Γ для произвольного значения τ вычисляются построчно через рекуррентные соотношения:

$$\gamma_{1k} = \tau^k/k! \quad (k = 1,\ldots,s+1); \quad \gamma_{jk} = (\tau - c_{j-1})\gamma_{j-1,k} - k\gamma_{j-1,k+1} \quad (j = 2,\ldots,s;\ k = 1,\ldots,s-j+2). \tag{34}$$

Нелинейные уравнения (31) относительно $\boldsymbol{u}_1,\ldots,\boldsymbol{u}_s,\boldsymbol{v}_1,\ldots,\boldsymbol{v}_s$ и $\boldsymbol{w}_1,\ldots,\boldsymbol{w}_s$ решаются на каждом шаге методом простых итераций в модификации Зейделя. До итераций известны решения $\boldsymbol{u}_1$, $\boldsymbol{v}_1$ и $\boldsymbol{w}_1$, а также разделенные разности

$$\boldsymbol{\alpha}_1 = \boldsymbol{f}_1 = \boldsymbol{f}(t_0,\boldsymbol{u}_1,\boldsymbol{v}_1,\boldsymbol{w}_1) \quad \text{и} \quad \boldsymbol{\beta}_1 = \boldsymbol{g}_1 = \boldsymbol{g}(t_0,\boldsymbol{u}_1,\boldsymbol{v}_1,\boldsymbol{w}_1)$$

на первой точке коллокаций $c_1 = 0$. В начале итерационного процесса на второй точке коллокаций c_2 определяется группа решений $\boldsymbol{u}_2$, $\boldsymbol{v}_2$ и $\boldsymbol{w}_2$, а из них – $\boldsymbol{f}_2$ и $\boldsymbol{g}_2$, по которым уточняются разделенные разности $\boldsymbol{\alpha}_2$ и $\boldsymbol{\beta}_2$ в соответствии с рекуррентными формулами (30). Затем таким же образом уточняются разделенные разности $\boldsymbol{\alpha}_3$ и $\boldsymbol{\beta}_3$ на третьей точке коллокаций c_3 и т.д. После поочередного уточнения всех разделенных разностей на шаге итерация повторяется. Итерационный процесс продолжается до тех пор, пока не выполнится неравенство

$$\|\boldsymbol{u}_s - \boldsymbol{u}_s^*\| < \varepsilon\,\|\boldsymbol{u}_s\|\,. \tag{35}$$

Здесь $\boldsymbol{u}_s^*$ – решение $\boldsymbol{u}_s$ на предыдущей итерации; ε – малая величина, которая задает точность сходимости. Впрочем, количество итераций на шаге можно ограничить заданным числом, не учи-

тывая выполнение условия (35). В таком случае нужно иметь в виду, что получаемое решение $\boldsymbol{u}_s$ уже не будет соответствовать задаваемому порядку, а метод лишится геометрических свойств.

В качестве начальных приближений разделенных разностей $\boldsymbol{\alpha}$ и $\boldsymbol{\beta}$ принимаются их оценки, получаемые из рекуррентных формул (30) по экстраполированным значениям функций $\boldsymbol{f}$ и $\boldsymbol{g}$:

$$\boldsymbol{f}_i = \boldsymbol{p}(1+c_i), \quad \boldsymbol{g}_i = \boldsymbol{r}(1+c_i) \quad (i=1,\ldots,s). \tag{36}$$

На первом шаге при отсутствии таких оценок итерационный процесс начинается буквально с нуля, т.е. при нулевых значениях разделенных разностей, и продолжается до обязательного выполнения условия (35).

При многократном выводе приближенных решений на плотной временной сетке удобно и целесообразно пользоваться подручным коллокационным полиномом, накрывающим на определенном шаге временные моменты сетки, вместо того чтобы выполнять пошаговое интегрирование на каждый из этих моментов с очень мелким шагом. Коллокационные полиномы для системы (28) на произвольный момент времени $t_0 + h\tau$ можно представить в виде

$$\boldsymbol{u}(\tau) = \boldsymbol{x}_0 + \boldsymbol{x}'_0 h\tau + h^2 \textstyle\sum_{j=1}^{s} \gamma_{j2}(\tau)\boldsymbol{\alpha}_j, \quad \boldsymbol{v}(\tau) = \boldsymbol{x}'_0 + h\textstyle\sum_{j=1}^{s} \gamma_{j1}(\tau)\boldsymbol{\alpha}_j, \quad \boldsymbol{w}(\tau) = \boldsymbol{z}_0 + h\textstyle\sum_{j=1}^{s} \gamma_{j1}(\tau)\boldsymbol{\beta}_j, \tag{37}$$

где величины $\gamma_{j1}(\tau)$ и $\gamma_{j2}(\tau)$ $(j=1,\ldots,s)$ вычисляются по формулам (34).

Альтернативный способ получения временных рядов приближенных решений – это полиномиальное интерполирование на промежуточных решениях, например, типа (29):

$$(\boldsymbol{u},\boldsymbol{v},\boldsymbol{w})(\tau) = \textstyle\sum_{j=1}^{s} (\boldsymbol{u}_j,\boldsymbol{v}_j,\boldsymbol{w}_j) \prod_{k=1}^{j-1} (\tau - c_k). \tag{38}$$

Однако точность интерполяционных полиномов (38) между точками коллокаций оказывается значительно ниже, чем точность коллокационных полиномов (37), получаемых прямым интегрированием полиномов (29).

Длина шага h как параметр интегратора задается пользователем. Однако возможен режим автоматического выбора длины шага в процессе пошагового интегрирования. Величина h выбирается таким образом, чтобы сохранялась приближенная оценка члена ряда Тейлора s-го порядка для вектора скорости [5]:

$$\| \boldsymbol{e} \|_{\text{cal}} = \frac{h}{s} \| \boldsymbol{\alpha}_s \| \approx \frac{h^s}{s!} \| \boldsymbol{x}'^{(s)} \|, \tag{39}$$

т.е. если рассматривать член ряда как главный член погрешности, длина шага будет определяться как для метода порядка $p = s-1$, хотя порядок на разбиениях гауссовых квадратур значительно больше (не менее чем в 2 раза). К сожалению, получить на шаге оценку более высокого порядка практически не удается.

Предположим, оценка (39) должна быть равна постоянной величине $\| \boldsymbol{e} \|_{\text{tol}}$, задаваемой пользователем. Поскольку

$$\| \boldsymbol{e} \|_{\text{cal}} \approx \frac{h^s}{s!} \| \boldsymbol{x}'^{(s)} \| \quad \text{и} \quad \| \boldsymbol{e} \|_{\text{tol}} \approx \frac{\hbar^s}{s!} \| \boldsymbol{x}'^{(s)} \|,$$

то требуемую для обеспечения величины $\| \boldsymbol{e} \|_{\text{tol}}$ длину шага $\hbar$ можно оценить как

$$\hbar = h \left(\frac{\| \boldsymbol{e} \|_{\text{tol}}}{\| \boldsymbol{e} \|_{\text{cal}}} \right)^{1/s}. \tag{40}$$

После получения на текущем шаге оценки (40) интегрирование не повторяется с новой длиной шага $\hbar$, но она используется для следующего шага по формуле

$$h_{n+1} = r h_n, \quad r = \left(\frac{s}{h_n} \frac{\| \boldsymbol{e} \|_{\text{tol}}}{\| \boldsymbol{\alpha}_s \|} \right)^{1/s}, \tag{41}$$

где n и $n+1$ – номера текущего и следующего шагов соответственно. Заметим, что автоматический выбор длины шага предполагает модификацию предиктора (36), а именно

$$\boldsymbol{f}_i = \boldsymbol{p}(1+rc_i), \quad \boldsymbol{g}_i = \boldsymbol{r}(1+rc_i) \quad (i=1,\ldots,s). \tag{42}$$

Алгоритм (41) должен быть эффективным, если h_n и h_{n+1} для любого n различаются несущественно. Заметные изменения в последовательности оценок (41) происходят, когда функция $\boldsymbol{f}$ ведет себя нерегулярно, например, при интегрировании вблизи ее сингулярностей. Чтобы избежать большого различия между h_n и h_{n+1}, к отношению $r = h_{n+1}/h_n$ следует применять демпфирование, т.е. наложить на него ограничения:

$$\sigma^{-1/s} < r < \sigma^{1/s}: \quad r < \sigma^{-1/s} \Rightarrow r = \sigma^{-1/s} \quad \text{или} \quad r > \sigma^{1/s} \Rightarrow r = \sigma^{1/s}. \tag{43}$$

Величина σ – это диапазон допустимого изменения значения $\| \boldsymbol{e} \|_{\text{cal}}$. Для изменения $\| \boldsymbol{e} \|_{\text{cal}}$ в пределах одного порядка $\sigma = \sqrt{10} \approx 3.16$. При невыполнении левого неравенства (43) шаг повторяется.

Начальный размер шага h_1 определяется из оценки (41) для $s = 2$ [5]:

$$h_1 = \sqrt{2\eta \frac{\| \boldsymbol{e} \|_{\text{tol}}}{\| \boldsymbol{f}_2 - \boldsymbol{f}_1 \|}}, \quad \boldsymbol{f}_1 = \boldsymbol{f}(t_0, \boldsymbol{x}_0, \boldsymbol{x}'_0, z_0), \quad \boldsymbol{g}_1 = \boldsymbol{g}(t_0, \boldsymbol{x}_0, \boldsymbol{x}'_0, z_0), \quad \boldsymbol{f}_2 = \boldsymbol{f}(t_1, \boldsymbol{x}_1, \boldsymbol{x}'_1, z_1),$$
$$\boldsymbol{x}_1 = \boldsymbol{x}_0 + \boldsymbol{x}'_0\eta + \tfrac{1}{2}\boldsymbol{f}_1\eta^2, \quad \boldsymbol{x}'_1 = \boldsymbol{x}'_0 + \boldsymbol{f}_1\eta, \quad z_1 = z_0 + \boldsymbol{g}_1\eta, \quad t_1 = t_0 + \eta. \tag{44}$$

Здесь η – некоторая малая величина. Если η мала настолько, что в компьютерной арифметике $\boldsymbol{f}_2 = \boldsymbol{f}_1$, то она увеличивается на порядок и оценка (44) выполняется снова.

Оценка стартового размера шага (44) в действительности адекватна только для метода Эйлера первого порядка. Поэтому для метода любого другого порядка стартовый шаг будет значительно меньше ожидаемого. Тем не менее пошаговый процесс интегрирования начинается с оценки (44), но в дальнейшем с использованием алгоритма (41) и с учетом ограничений (43) размер шага постепенно будет выходить на должную величину штатного режима работы интегратора с автоматическим выбором длины шага.

Последний шаг выявляется из условия

$$(t_0 + \Delta t - t_n)/h_{n+1} < 1, \tag{45}$$

где Δt – длина всего интервала интегрирования; t_n – момент времени на n-м шаге. Тогда, чтобы выйти на конечный момент времени $t_0 + \Delta t$, размер последнего шага задается принудительно как $h_{n+1} = t_0 + \Delta t - t_n$ и переопределяется его отношение к длине текущего шага h_n: $r = h_{n+1}/h_n$.

3. Процедура Lobbie

Интегратор реализован на процедурном языке Фортран до 32-го порядка для компьютерной арифметики с двойной и четверной точностью (double & quadruple precision). Вызов программной процедуры интегратора Lobbie выполняется командой

call lobbie(x, y, z, ts, tf, step, etol, nxy, nz, ns, ni, nst, ncf, fun).

Здесь **x**, **y**, **z** – массивы интегрируемых переменных $\boldsymbol{x}, \boldsymbol{x}', z$ соответственно: на входе значения на начальный момент времени t_0 (**ts**), на выходе значения на конечный момент времени $t_0 + \Delta t$ (**tf**); **step** – начальный размер шага интегрирования h_1: при автоматическом выборе (41) на выходе значение h_n (размер предпоследнего шага), при нулевом значении **step** величина h_1 задается интегратором, согласно оценке (44); **etol** – $\| \boldsymbol{e} \|_{\text{tol}}$: при нулевом значении – режим постоянного шага интегрирования; **nxy** и **nz** – размерности массивов **x**, **y** ($\dim \boldsymbol{x} = \dim \boldsymbol{x}'$) и **z** ($\dim z$) соответственно; **ns** – количество узловых значений s; **ni** – максимальное число итераций на шаге для решения нелинейных уравнений (31) относительно $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_s, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_s$ и $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_s$; **nst** и **ncf** – количество выполненных шагов и обращений к процедуре **fun** для вычисления функций $\boldsymbol{f}$ и $\boldsymbol{g}$ на всем интервале интегрирования. Процедура **fun** задается как

subroutine fun(t, x, y, z, f),

где **t** – текущий момент времени t; **x**, **y**, **z** – массивы интегрируемых переменных со значениями на момент t; **f** – выходной массив значений функций $\boldsymbol{f}$ и $\boldsymbol{g}$ размерности $\dim \boldsymbol{f} + \dim \boldsymbol{g}$.

Вычислительный процесс внутри процедуры Lobbie выполняется в соответствии с блок-схемой, представленной на рис. 2. Опишем коротко основные этапы пошагового интегрирования в

случае переменного шага с автоматическим выбором его стартовой величины, а также при возрастающем изменении времени $(\Delta t > 0)$.

I. Из прилагаемого к процедуре блока данных считывается массив узловых значений $c_1,\ldots,c_s$ и рекуррентно вычисляются константы интегратора (32) с использованием (34). Оценивается стартовая величина шага h_1 (44). Из выполнения условия $h_1 > \Delta t$ устанавливается, что стартовый шаг является последним, и тогда задается величина $h_1 = \Delta t$.

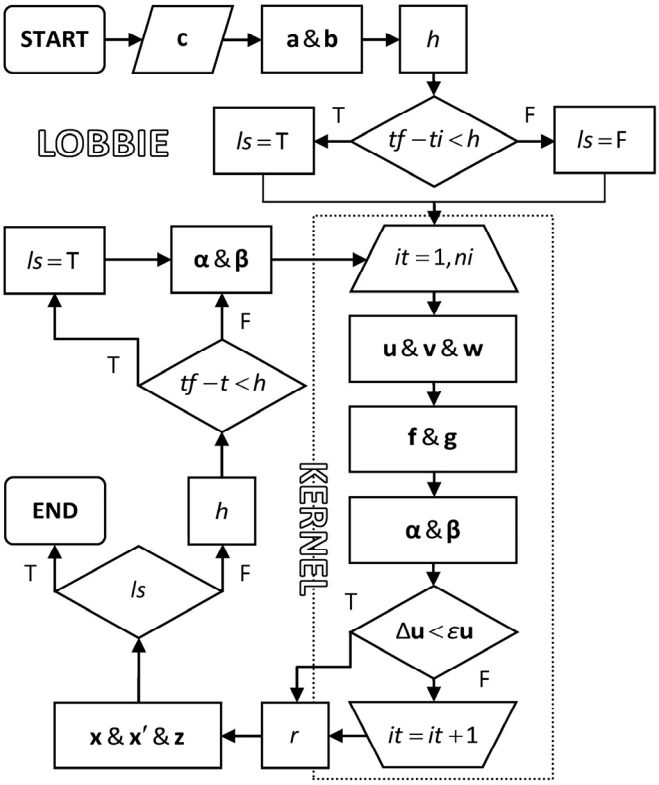

Рис. 2. Блок-схема программной процедуры Lobbie

II. Итерационно определяются промежуточные решения $\boldsymbol{u}_1,\ldots,\boldsymbol{u}_s, \boldsymbol{v}_1,\ldots,\boldsymbol{v}_s$ и $\boldsymbol{w}_1,\ldots,\boldsymbol{w}_s$ (31), а также функции $\boldsymbol{f}_1,\ldots,\boldsymbol{f}_s$ и $\boldsymbol{g}_1,\ldots,\boldsymbol{g}_s$ (28) вместе с разделенными разностями $\boldsymbol{\alpha}_1,\ldots,\boldsymbol{\alpha}_s$ и $\boldsymbol{\beta}_1,\ldots,\boldsymbol{\beta}_s$. Итерации завершаются при выполнении условия (35) или по достижении максимально допустимого количества итераций ni.

III. После итерационного процесса формируются решения на шаге $\boldsymbol{x}_1, \boldsymbol{x}'_1, \boldsymbol{z}_1$ (31) с учетом уточненных разделенных разностей $\boldsymbol{\alpha}_s$ и $\boldsymbol{\beta}_s$ на последней итерации. Оценивается масштабирующий множитель r (41) с учетом демпфирующих условий (43).

IV. Если шаг последний, процедура завершается. Иначе определяется величина следующего шага (41). Выполнение условия (45) устанавливает последний шаг интегрирования и тогда его величина переопределяется так, чтобы обеспечить выход на конечный момент времени $t_0 + \Delta t$.

V. Экстраполируются значения разделенных разностей (30) с использованием значений функций дифференциальных уравнений (42) и вычислительный процесс повторяется от этапа II, где полученные решения $\boldsymbol{x}_1, \boldsymbol{x}'_1, \boldsymbol{z}_1$ принимаются за начальные $\boldsymbol{x}_0, \boldsymbol{x}'_0, \boldsymbol{z}_0$.

Заключение

Таким образом, в статье изложены теоретические основы предлагаемого нового коллокационного интегратора для решения смешанных систем дифференциальных уравнений динамики первого и второго порядков. Акцентированы особенности в практической реализации интегратора,

а также описана его программная процедура Lobbie. В дальнейших работах будут представлены результаты тестирования нового интегратора на примере простых динамических систем, а также показана его эффективность в сравнении с другими широко используемыми интеграторами.

СПИСОК ЛИТЕРАТУРЫ

1. Guillou A. and Soule J.L. // Rev. Francaise Informat. Recherche Oprationnelle 3. – 1969. – Ser. R-3. – P. 17–44.
2. Wright K. // BIT. – 1970. – V. 10. – P. 217–227.
3. Hairer E., Norsett S.P., Wanner G. Solving Ordinary Differential Equations. Nonstiff Problems. – Springer, 2008. – 528 p.
4. Hairer E., Lubich C., Wanner G. Geometric Numerical Integration. Structure-Preserving Algorithms for Ordinary Differential Equations. – Springer, 2006. – 644 p.
5. Авдюшев В.А. Численное моделирование орбит небесных тел. – Томск: Издат. Дом Томского государственного университета, 2015. – 336 с.
6. Everhart E. // Celest. Mech. – 1974. – V. 10. – P. 35–55.
7. Everhart E. // Dynamics of Comets: Their Origin and Evolution. Proceedings of IAU Colloq. 83, held in Rome, Italy, June 11–15, 1984 / ed. by A. Carusi and G. B. Valsecchi. – Dordrecht: Reidel, Astrophysics and Space Science Library, 1985. – V. 115. – P. 185–202.
8. Kuntzmann J. // Z. Angew. Math. Mech. – 1961. – V. 41. – P. 28–31.
9. Butcher J.C. // Math. Comput. – 1964. – V. 18. – P. 50–64.
10. Cincotta P.M., Giordano C.M., and Simó C. // Physica D. – 2003. – V. 182. – P. 151–178.
11. Burdet C.A. // Z. Angew. Math. Phys. – 1968. – V. 19. – P. 345–368.
12. Kustaanheimo P. and Stiefel E. // J. Reine Angew. Math. – 1965. – V. 218. – P. 204–219.
13. Шефер В.А. // Астрон. журн. – 1991. – Т. 68. – С. 197–205.
14. Baumgarte J. // Comp. Math. Appl. Mech. Eng. – 1972. – V. 1. – P. 1–16.

Поступила в редакцию 28.04.2020.

Национальный исследовательский Томский государственный университет,
г. Томск, Россия

Авдюшев Виктор Анатольевич, д.ф.-м.н., профессор НИ ТГУ, e-mail: scharmn@mail.ru.