**КОЛЛОКАЦИОННЫЙ ИНТЕГРАТОР,
ПОСТРОЕННЫЙ С ПОМОЩЬЮ ПОЛИНОМОВ ЛЕЖАНДРА**

В. Ш. Шайдулин, А. С. Русаков
Санкт-Петербургский государственный университет

В данной работе представлена реализация коллокационного интегратора для систем обыкновенных дифференциальных уравнений первого порядка. На каждом шаге строится интерполяционный полином, который представляется линейной комбинацией по полиномам Лежандра. Такое представление позволяет построить эффективный алгоритм вычисления коэффициентов интерполяционного полинома и результата интегрирования на каждом шаге. Точность интегратора определяется набором промежуточных узлов, выбираемых для построения интерполяционного полинома.

COLLOCATION INTEGRATOR BASED ON LEGENDRE POLYNOMIALS

V. Sh. Shaidulin, A. S. Rusakov
Saint-Petersburg State University

This paper presents the implementation of a collocation integrator for system of first-order ordinary differential equations. At each step, we construct the interpolation polynomial represented by a linear combination of the Legendre polynomials. Such a representation allows us to make an effective algorithm for calculating the coefficients of the interpolation polynomial and the result of integration at each step. The accuracy of the integrator is determined by the set of intermediate nodes selected to construct the interpolation polynomial.

Введение

Коллокационные методы интегрирования представляют собой иную интерпретацию ряда методов класса Рунге—Кутта, которую впервые заметили Хаммер и Холлинсворт [1]. В дальнейшем удалось обобщить идею представления решения в полиномиальном виде для методов разных порядков [2]. Интересно отметить, что, в отличие от других методов интегрирования, мы получаем непрерывное решение на каждом шаге. В данной работе представлена одна из возможных реализаций коллокационного метода интегрирования.

Реализация коллокационного метода интегрирования

Допустим, что нам необходимо проинтегрировать систему обыкновенных дифференциальных уравнений

$$\dot{\mathbf{y}} = \mathbf{f}(t, \mathbf{y}), \tag{1}$$

где $\mathbf{y}$ — вектор некой размерности n; $\mathbf{f}$ — вектор-функция правых частей и t — время. Коллокационные методы интегрирования предлагают представлять решение системы на шаге величины h с началом в t_0 в виде полинома $\mathbf{u}$ заданной степени s:

$$\mathbf{y}(t_0 + h\tau) \approx \mathbf{u}(\tau).$$

Величина τ представляет собой безразмерное время, изменяющееся на отрезке $[0, 1]$.

Полином $\mathbf{u}$ мы можем представить различными способами, нам удобен следующий вид:

$$\mathbf{u}(\tau) = \sum_{k=0}^{s} \boldsymbol{\alpha}_k P_k(\tau).$$

Здесь $\boldsymbol{\alpha}_k$ — векторные коэффициенты полинома, разложенного по базису смещенных полиномов Лежандра $P_k(\tau)$, определенных на отрезке $[0, 1]$:

$$P_k(\tau) = \frac{1}{k!} \frac{d^k}{d\tau^k} \Big(\tau^k (\tau - 1)^k \Big).$$

Если получится каким-либо образом определить значение всех коэффициентов $\boldsymbol{\alpha}_k$, то мы получим оценку $\mathbf{y}$ в момент времени $t_0 + h$:

$$\mathbf{y}(t_0 + h) \approx \mathbf{u}(1) = \sum_{k=0}^{s} \boldsymbol{\alpha}_k P_k(1) = \boldsymbol{\alpha}_0 + \sum_{k=1}^{s} \boldsymbol{\alpha}_k. \tag{2}$$

Последнее равенство выполняется, потому что $P_k(1) = 1$ для любого k.

Для поиска значений $s + 1$ коэффициента $\boldsymbol{\alpha}_k$ можно составить систему уравнений с помощью (1). Вначале заметим, что

$$\dot{\mathbf{y}} h \approx \mathbf{u}'. \tag{3}$$

Здесь точкой обозначается производная по времени t, а штрихом — по безразмерному времени τ. Мы можем определить s уникальных точек $c_j \in [0, 1]$ и для каждой из них с помощью (1) и (3) записать уравнение:

$$\sum_{k=1}^{s} \boldsymbol{\alpha}_k P'_k(c_j) = \mathbf{f}(t_0 + hc_j, \mathbf{y}_j) h, \qquad \mathbf{y}_j = \sum_{k=0}^{s} \boldsymbol{\alpha}_k P_k(c_j). \tag{4}$$

В итоге составляется система из s нелинейных уравнений, и, чтобы она была полной, требуется добавить еще одно уравнение:

$$\mathbf{y}(t_0) = \boldsymbol{\alpha}_0 + \sum_{k=1}^{s} \boldsymbol{\alpha}_k P_k(0) = \boldsymbol{\alpha}_0 + \sum_{k=1}^{s} \boldsymbol{\alpha}_k (-1)^k. \tag{5}$$

Решать систему можно итеративными способами, начиная с некоторого начального приближения. Например, вначале можно вычислять правую часть (4), используя имеющиеся $\boldsymbol{\alpha}_k$, разрешать (4) как систему линейных уравнений для $\boldsymbol{\alpha}_k$ при $k = 1, \ldots, s$ (метод Пикара) и из (5) определять $\boldsymbol{\alpha}_0$. Эту процедуру можно повторять до достижения необходимой точности.

Порядок точности построенного интегратора определяется узлами сетки c_j и совпадает с порядком точности квадратурной формулы интерполяционного типа, построенной на той же сетке. Например, мы можем использовать квадратуру Лобатто и назначить узлами сетки корни интеграла от смещенного полинома Лежандра степени $s - 1$:

$$\int_0^{\tau} P_{s-1}(\tau)\, d\tau = \frac{1}{(s-1)!} \frac{d^{s-2}}{d\tau^{s-2}} \Big(\tau^{s-1} (\tau - 1)^{s-1} \Big).$$

Тогда точность получаемого интегратора $O(h^{2s-2})$.

Альтернативно мы можем рассмотреть систему (4) как матричное уравнение, если соберем $\boldsymbol{\alpha}_k$ при $k = 1, \ldots, s$ в матрицу $\mathcal{A}$ размера $n \times s$, $\mathbf{f}(t_0 + hc_j, \mathbf{y}_j)h$ в матрицу $\mathcal{F}$

размера $n \times s$ и $P'_k(c_j)$ в матрицу $\mathcal{C}$ размера $s \times s$. В матрицах $\mathcal{A}$ и $\mathcal{F}$ столбцы представляют соответствующие векторы в порядке возрастания индексов k и j. В матрице $\mathcal{C}$ индекс k перечисляет столбцы, а j — строки. Тогда мы можем записать (4) в виде

$$\mathcal{A}\mathcal{C}^T = \mathcal{F}.$$

Теперь на каждом шаге итеративного процесса мы можем вычислять напрямую

$$\mathcal{A} = \mathcal{F}\mathcal{C}^{-T}, \tag{6}$$

поскольку $\mathcal{C}^{-T}$ нам известна заранее и не требует повторных вычислений.

К тому же мы можем исключить из алгоритма вычисление коэффициента $\boldsymbol{\alpha}_0$, если выразим его из (5):

$$\boldsymbol{\alpha}_0 = \mathbf{y}(t_0) - \sum_{k=1}^{s} \boldsymbol{\alpha}_k (-1)^k.$$

Подставляя полученное выражение в (2), получим

$$\mathbf{y}(t_0 + h) = \mathbf{y}(t_0) - \sum_{k=1}^{s} \boldsymbol{\alpha}_k (-1)^k + \sum_{k=1}^{s} \boldsymbol{\alpha}_k = \mathbf{y}(t_0) + \sum_{k=1}^{s} \boldsymbol{\alpha}_k \Big(1 - (-1)^k\Big). \tag{7}$$

В последней сумме слагаемые с четными k пропадают. Вычисление $\mathbf{y}_j$ в (4) и (6) меняется следующим образом:

$$\mathbf{y}_j = \mathbf{y}(t_0) + \sum_{k=1}^{s} \boldsymbol{\alpha}_k \Big(P_k(c_j) - (-1)^k\Big) = \mathbf{y}(t_0) + \mathcal{A}\mathcal{B}_j, \tag{8}$$

где $\mathcal{B}_j$ — вектор-столбец размера s, составленный из величин в скобках в последней сумме.

Заключение

Приведенная схема интегрирования представляет общий шаблон для построения интеграторов разной точности и свойств, которые наследуются от квадратурных сеток, используемых для нахождения коэффициентов α_k. Например, квадратура Лобатто позволяет построить симметричный интегратор высокой точности, который сохраняет ряд интегралов дифференциальных уравнений, связанных с симметрией по времени (если таковые существуют). Код интегратора реализован на языке программирования `C++` (стандарт `C++20`) и доступен для скачивания по ссылке `https://github.com/shvak/collo`

Работа выполнена при поддержке гранта РНФ 19-72-10023.

Библиографические ссылки

[1] *Hammer P. C., Hollingsworth J. W.* Trapezoidal methods of approximating solutions of differential equations // Math. Tables Aids Comput. — 1955. — Vol. 9. — P. 92—96.

[2] *Hairer E., Wanner G., Lubich C.* Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations. Second Edition. — Springer, 2006. — P. 644.