Обзор методов и инструментальных средств решения задачи Коши для обыкновенных дифференциальных уравнений с гарантированной оценкой погрешности

А.В. Позин
Пермский государственный университет
e-mail: alexey.pozin@gmail.com

Дифференциальные уравнения широко используются для решения различных классов прикладных задач, к примеру, различных задач в физике и экономике. Хотя некоторые простейшие типы дифференциальных уравнений имеют аналитическое решение, большая часть реальных прикладных задач, как правило, описывается сложными системами дифференциальных уравнений, не имеющими аналитического решения.

Недостатком классических методов численного решения задачи Коши для ОДУ является то, что они в общем случае не гарантируют попадания приближённого решения в предполагаемый интервал погрешности. Методы решения задачи Коши для ОДУ с гарантированной оценкой погрешности (или интервальные методы), наоборот, изначально предполагают решение систем ОДУ в виде интервала таким образом, что точное решение гарантированно попадает в полученный интервал.

В данном обзоре мы будем рассматривать методы и инструментальные средства решения задачи Коши для систем обыкновенных дифференциальных уравнений с гарантированной оценкой погрешности.

1. Общие понятия

Рассмотрим задачу Коши для системы автономных обыкновенных дифференциальных уравнений с начальными условиями в виде интервалов [1]:

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

$$y(t_0) \in [y_0], \tag{2}$$

где $t \in [t_0, T],\ T > t_0$. Здесь t_0 и $T \in \mathbb{R}$, $f \in C^{k-1}(\mathcal{D}),\ k \geq 2$, $\mathcal{D} \subseteq \mathbb{R}^n$ – открытое подмножество, $f : \mathcal{D} \to \mathbb{R}^n$ и $[y_0] \subseteq \mathcal{D}$.

Далее определим разбиение на временной оси $t_0 < t_1 < \ldots < t_m = T$, размеры интервалов разбиения в общем случае не равны. Размер шага от t_j до t_{j+1} обозначим за h_j ($h_j = t_{j+1} - t_j$). Шаг от t_j до t_{j+1} обозначается как $(j+1)$-й шаг.

Обозначим решение задачи с начальными условиями y_j в точке t_j за $y(t; t_j, y_j)$. Для интервала, или, в общем случае, интервального вектора, обозначим за $y(t; t_j, [y_j])$ множество решений

$$\left\{ y\left(t; t_j, y_j\right) \mid y_j \in [y_j] \right\}.$$

Целью является вычисление интервальных векторов $[y_j]$, $j = 1, 2, \ldots, m$, которые гарантированно содержат решение исходной задачи в узлах сетки разбиения. Таким образом,

$$y\left(t_j; t_0, [y_0]\right) \subseteq [y_j],\ j = 1, 2, \ldots, m.$$

Классические численные методы решения задачи Коши для ОДУ вычисляют приближённое решение, которое удовлетворяет определённой пользователем точности [2]. Эти методы обычно просты в реализации и надёжно решают большую часть задач, однако, в некоторых случаях возвращают некорректные результаты. Методы с гарантированной оценкой погрешности (или интервальные методы) наоборот требуют более сложных вычислений. Но при этом, если метод с гарантированной оценкой погрешности смог получить решение задачи, то это решение не только гарантированно содержит точное решение, но и является доказательством существования и единственности точного решения [3].

Ранее, интервальные методы не пользовались популярностью из-за высоких требований к вычислительным ресурсам, по сравнению с классическими методами. Однако сейчас, при современной производительности вычислительной техники, методы с гарантированной оценкой погрешности становятся всё более востребованными. Сейчас всё чаще встречаются прикладные задачи, в которых требуется получить интервал, гарантированно содержащий точное решение.

Кроме того, в некоторых случаях интервальные методы решения задачи Коши для ОДУ не требуют больших вычислительных ресурсов по сравнению с классическими методами. К примеру, множество прикладных задач требуют решения параметрических ОДУ, параметры которых можно оценить только в виде некоторого диапазона значений. Для таких задач интервальный метод может получить решение сразу для всего диапазона значений, не прибегая к дополнительным вычислениям.

2. Метод интервальных рядов Тейлора

2.1. Общая схема метода

Метод интервальных рядов Тейлора, как и большинство других интервальных методов решения задачи Коши для ОДУ, заключается в разбиении исходного периода интегрирования на определённое число шагов, при этом каждый шаг интегрирования состоит из 2 этапов:

1. **Алгоритм 1.** На этом этапе вычисляется размер текущего шага интегрирования h_j и априорный интервал $[\tilde{y}_j]$, гарантированно содержащий единственное решение $y(t; t_j, y_j)$ для всех $t \in [t_j, t_{j+1}]$ и $y_j \in [y_j]$. Таким образом,
$$y(t; t_j, [y_j]) \subseteq [\tilde{y}_j], \qquad \text{для всех } t \in [t_j, t_{j+1}].$$
Как правило, алгоритм для проверки существования и единственности решения на отрезке $[t_j, t_{j+1}]$ использует оператор Пикара-Линделёфа и теорему Банаха о неподвижной точке.

2. **Алгоритм 2.** На этом этапе с помощью априорного интервала $[\tilde{y}_j]$ вычисляется более узкий интервал $[y_{j+1}]$, гарантированно содержащий решение исходной задачи в точке t_{j+1}:
$$y(t_{j+1}; t_0, [y_0]) \subseteq [y_{j+1}].$$
В методе интервальных рядов Тейлора для вычисления более точного приближения используются разложение в ряд Тейлора с остатком, теорема о среднем и различные интервальные преобразования.

Далее рассмотрим каждый из алгоритмов более подробно.

2.2. Алгоритм 1

Алгоритм 1 используется для получения априорных оценок необходимого размера шага и интервала, гарантированно содержащего решение задачи (1) с начальными условиями (2). Метод постоянного приближения, используемый в алгоритме 1, предполагает применение оператора Пикара-Линделёфа к определённому набору функций. Оператор Пикара-Линделёфа выглядит следующим образом:

$$(Ty)(t) = y_j + \int\limits_{t_j}^{t} f(y(s))ds.$$

Теорема Банаха, также используемая в методе постоянного приближения, гарантирует наличие и единственность неподвижной точки у некоторых отображений метрических пространств. Также она содержит конструктивный метод нахождения этой точки. Теорема звучит следующим образом:

Пусть $(\mathbb{X}, d)$ - непустое полное метрическое пространство. Пусть $T : \mathbb{X} \mapsto \mathbb{X}$ – сжимающее отображение на $\mathbb{X}$, то есть существует число $0 \leq \alpha < 1$ такое, что

$$d(Tx, Ty) \leq \alpha d(x, y),$$

для всех $x, y \in \mathbb{X}$. Тогда у отображения T существует, и притом ровно одна неподвижная точка $x^* \in \mathbb{X}$, таким образом $Tx^* = x^*$. Число α часто называют коэффициентом сжатия.

Для реализации алгоритма 1 часто применяют метод постоянного приближения. Метод постоянного приближения первого порядка для автономных задач Коши звучит следующим образом:

Если h_j и $[\tilde{y}_j]$ удовлетворяют условию:

$$\left[\tilde{y}_j^1\right] = [y_j] + [0, h_j]\, f([\tilde{y}_j]) \subseteq [\tilde{y}_j]\,,$$

то задача Коши имеет единственное решение $y(t; t_j, y_j) \subseteq \left[\tilde{y}_j^1\right]$ для всех $t \in [t_j, t_{j+1}]$ и $y_j \in [y_j]$.

Метод постоянного приближения первого порядка достаточно просто реализуем, но серьёзным недостатком является то, что данный метод часто предъявляет слишком строгие ограничения на размер отрезка разбиения h_j. Для того, чтобы избавиться от этого недостатка, можно использовать метод постоянного приближения высокого порядка, основанный на теореме Корлисса и Рима [4]. Этот метод требует более сложных вычислений и большей производительности вычислительной системы, но при этом позволяет использовать более крупные отрезки разбиения h_j.

2.3. Алгоритм 2

Рассмотрим разложение функции в ряд Тейлора:

$$y_{j+1} = y_j + \sum_{i=1}^{k-1} h_j^i f^{[i]}(y_j) + h_j^k f^{[k]}(y; t_j, t_{j+1}),$$

где $y_j \in [y_j]$ и $f^{[k]}(y; t_j, t_{j+1})$ обозначает функцию $f^{[k]}$, l-й компонент которой ($l = 1, \ldots, n$) вычислен в точке $y(\xi_{jl})$, для некоторого $\xi_{jl} \in [t_j, t_{j+1}]$.

Используя априорную оценку $[\tilde{y}_j]$ на отрезке $[t_j, t_{j+1}]$ можно оценить разложение Тейлора следующим образом:

$$[y_{j+1}] = [y_j] + \sum_{i=1}^{k-1} h_j^i f^{[i]}([y_j]) + h_j^k f^{[k]}([\tilde{y}_j]),$$

при этом данное разложение будет содержать решение $y(t_{j+1}; t_0, [y_0])$ задачи Коши для системы автономных дифференциальных уравнений.

При реализации метода интервальных рядов Тейлора зачастую возникает проблема снижения точности на каждом шаге, связанная с так называемым эффектом обёртывания [1, с. 134]. Для снижения эффекта обёртывания применяются различные методы. Одним из самых эффективных методов снижения эффекта обёртывания является метод QR-факторизации Лонера [5].

3. Реализации метода рядов Тейлора

Способы реализации интервальных методов решения задачи Коши на данный момент изучены гораздо меньше, чем для классических методов. К примеру, всё ещё довольно слабо изучены стратегии управления размером шага и порядком методов, а также методы для решения жестких задач. Кроме того, до сих пор остаются открытыми задачи уменьшения эффекта обёртывания в алгоритме 2 и использования большего размера шага в алгоритме 1. С учётом всех используемых для расчета инструментов, интервальный решатель должен быть более сложным в реализации, чем классический решатель для ОДУ. Помимо библиотеки интервальной арифметики, одним из ключевых компонентов интервального решателя является библиотека для автоматического вычисления интервальных коэффициентов Тейлора.

Можно выделить 5 основных библиотек для решения задачи Коши для обыкновенных дифференциальных уравнений с гарантированной оценкой погрешности [6]:

- AWA;
- ADIODES;
- COSY INFINITY;
- VNODE;
- VSPODE.

Рассмотрим кратко каждую из этих библиотек.

Библиотека AWA [7] – это библиотека для решения ОДУ с гарантированной оценкой погрешности, разработанная Р. Лонером. Данная библиотека представляет собой реализацию метода Лонера в качестве алгоритма 2 и метода постоянного приближения для проверки существования и единственности решения. Библиотека AWA доступна для скачивания по ссылке http://www2.math.uni-wuppertal.de/∼xsc/xsc/pxsc_software.html #awa.

Библиотека ADIODES [8] – это C++ реализация решателя с использованием метода постоянного приближения в алгоритме 1 и метода Лонера в алгоритме 2. Размер шага и в AWA и в ADIODES часто ограничен шагами Эйлера, получаемыми в результате выполнения алгоритма 1. Библиотека ADIODES доступна для скачивания по ссылке http://www.imm.dtu.dk/fadbad.html#download.

Библиотека COSY INFINITY [9] – это библиотека, реализованная на языке Fortran и предназначенная для проектирования и изучения систем лазерной физики. Реализованный в данной библиотеке интервальный метод решения ОДУ использует полиномы Тейлора высоких порядков по времени и начальным условиям. Эффект обёртывания снижается за счёт установки функциональных зависимостей между начальными и конечными условиями. Для этого все вычисления с коэффициентами полиномов Тейлора выполняются на множестве вещественных чисел, а интервальные границы вычисляются только для остатка. При данном подходе, все арифметические операции и стандартные функции должны работать с целыми полиномами в качестве операндов. Хотя такой подход довольно эффективно снижает эффект обёртывания, работа с полиномами выполняется намного медленнее работы с интервалами.

Библиотека VNODE (Validated Numerical ODE) [3] – это объектно-ориентированная библиотека, реализованная на языке C++ и использующая метод постоянного приближения высокого порядка в качестве алгоритма 1 и интервальный метода Эрмита-Обрешкова [10] в качестве алгоритма 2. Библиотека VNODE может использовать интервальные параметры и интервальные начальные условия в качестве входных данных. Для реализации интервальной арифметики и автоматического расчета коэффициентов Тейлора используются библиотеки PROFIL/BIAS [11] и FADBAD/TADIFF [12, 13] соответственно. Среди достоинств библиотеки можно выделить довольно прозрачную внутреннюю структуру, упрощающую понимание реализации и последующие модификации библиотеки. Одним из недостатков библиотеки VNODE является то, что при использовании интервальных параметров в системе ОДУ, результаты метода довольно часто начинают сильно расходиться уже на ранних шагах, что не позволяет получить корректное решение на заданном отрезке. Кроме того, библиотека не всегда может решить сложные нелинейные задачи.

Библиотека VSPODE (Validating Solver for Parametric ODEs) [14] – это библиотека, написанная на языке C++ и являющаяся своеобразным расширением библиотеки VNODE для более эффективного расчета задач Коши с интервальными параметрами. В качестве алгоритма 1 библиотека VSPODE использует метод постоянного приближения высокого порядка. В качестве алгоритма 2 библиотека использует интервальные ряды Тейлора, для которых предварительно выполняются определённые преобразования над интервальными параметрами, что позволяет значительно снизить эффект обёртывания и получать более адекватные результаты для параметрических ОДУ. Библиотека использует пакет FADBAD/TADIFF для автоматического расчета коэффициентов Тейлора, и собственную реализацию функций интервальной арифметики.

Список литературы

[1] Moore R. Interval analysis. N.J.: Prentice-Hall, 1966.

[2] Lee H.J., Schiesser W.E. Ordinary and partial differential equation routines in C, C++, Fortran, Java, Maple, and MATLAB. London: Chapman and Hall/CRC, 2004.

[3] Nedialkov N.S. The design and implentation of an object-oriented validated ODE Solver // available at http://www.cs.toronto.edu/NA/reports.html#ned.software.01.

[4] Nedialkov N.S., Jackson K.R., Pryce J.D. An Effective High-Order Interval Method for Validating Existence and Uniqueness of the Solution of an IVP for an ODE // Reliable Computing. 2001. Vol. 7, N 6. P. 449–465.

[5] Lohner R.J. Enclosing the solutions of ordinary initial and boundary value problems // Computer Arithmetic: Scientific Computation and Programming Languages. 1987. P. 255–286.

[6] Nedialkov N.S., Jackson K.R. ODE software that computes guaranteed bound on the solution // available at http://www.cs.torono.edu/NA/reports.html#SciTools.98.

[7] Lohner R.J. Einschliessung der Lösung gewöhnlicher Anfangsund Randwertaufgaben und Anwendungen. PhD thesis, Universität Karlsruhe, 1988.

[8] Stauning O. Automatic Validation of Numerical Solutions. PhD thesis, Technical University of Denmark, Lyngby, Denmark, October 1997. // available at http://www.imm.dtu.dk/fadbad.html#download.

[9] Berz M. COSY INFINITY version 8 reference manual. Technical Report MSUCL–1088, National Superconducting Cyclotron Lab., Michigan State University, East Lansing, Mich., 1997. // available at http://www.beamtheory.nscl.msu.edu/cosy/.

[10] Nedialkov N.S., Jackson K.R. An interval HermiteObreschkoff method for computing rigorous bounds on the solution of an initial value problem for an ordinary differential equation // Reliable Computing. 1999. Vol. 5, N 3. P. 289–310.

[11] Knüppel O. PROFIL/BIAS – a fast interval library // Computing. 1994. Vol. 5, N 3. P 277–287.

[12] Bendsten C., Stauning O. FADBAD, a flexible C++ package for automatic differentiation using the forward and backward methods. Technical Report 1996x594, Department of Mathematical Modelling, Technical University of Denmark, DK2800, Lyngby, Denmark, 1996. // available at http://www.imm.dtu.dk/fadbad.html.

[13] Bendsten C., Stauning O. TADIFF, a flexible C++ package for automatic differentiation using Taylor series. Technical Report 1997x594, Department of Mathematical Modelling, Technical University of Denmark, DK2800, Lyngby, Denmark, 1997. // available at http://www.imm.dtu.dk/fadbad.html.

[14] Lin Y., Stadtherr M.A. Validated solutions of initial value problems for parametric ODEs // Applied Numerical Mathematics. 2007. Vol. 57, N 10. P. 1145–1162.