Высокоточные вычисления для моделирования нелинейных динамических систем с аттракторами

А. Н. Пчелинцев, email: pchelintsev.an@yandex.ru [1]

[1] ФГБОУ ВО «Тамбовский государственный технический университет»

Аннотация. *В данном докладе рассматриваются метод, алгоритм и программная реализация, использующие высокоточные вычисления для построения приближений к неустойчивым решениям динамических систем с квадратичной правой частью в их аттракторах. Частный случай таких систем – система Лоренца. Описаны используемые библиотеки в разработанном комплексе программ.*

Ключевые слова: *аттрактор, высокоточные вычисления, степенной ряд, MPFR C++, Boost uBLAS.*

Введение

Рассмотрим динамическую систему с квадратичными нелинейностями вида

$$\dot{X} = B_0 + B_1 X + \Phi(X), \tag{1}$$

где $X(t) = [x_1(t) \ldots x_n(t)]^{\mathrm{T}}$ – векторная функция времени t со значениями в пространстве R^n, $B_0 \in \mathrm{R}^n$ – заданный вектор-столбец,

$$\Phi(X) = [\varphi_1(X) \ldots \varphi_n(X)]^{\mathrm{T}},$$

$\varphi_p(X) = \left\langle Q_p X, X \right\rangle$, B_1 и Q_p ($p = 1, \ldots, n$) – матрицы $n \times n$ действительных чисел.

Активный интерес в научной литературе к системам вида (1) возник, начиная где-то с середины XX века. Самый популярный объект для компьютерного моделирования из класса систем (1) – это система Лоренца [1]:

$$\begin{cases} \dot{x}_1 = \sigma(x_2 - x_1), \\ \dot{x}_2 = r x_1 - x_2 - x_1 x_3, \\ \dot{x}_3 = x_1 x_2 - b x_3. \end{cases} \tag{2}$$

Для системы (2) матрицы имеют вид:

$$B_0 = \bar{0},\; B_1 = \begin{bmatrix} -\sigma & \sigma & 0 \\ r & -1 & 0 \\ 0 & 0 & -b \end{bmatrix},\; Q_1 = O,\; Q_2 = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

$$Q_3 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

Подобные системы представляют интерес для исследователей, поскольку они являются системами с рассеиванием энергии, т.е. диссипативными. В них присутствует сжатие объема фазового пространства, и все траектории равномерно ограничены. Таким образом, существует притягивающее множество, к которому стремятся все решения при $t \to \infty$, называемое аттрактором динамической системы. Следовательно, аттракторы диссипативных систем определяют их поведение на больших отрезках времени. Однако для многих систем вида (1) предельные решения неустойчивы. Поэтому для корректного численного интегрирования нужны высокоточные методы, позволяющие гибко управлять накоплением ошибки вычислений на большом отрезке времени. Например, с повышением порядка методов Рунге-Кутты растет сложность формул вычисления приближенных значений фазовых координат (они имеют древовидную структуру) [2, 3].

В последнее время для моделирования неустойчивых предельных решений систем вида (1) многие исследователи (например, [4, 5]) начали использовать эффективные модификации метода степенных рядов.

В данной работе будут рассмотрены метод, алгоритм и программная реализация, использующие высокоточные вычисления для построения приближений к неустойчивым решениям системы (1).

2. Разложение решений в степенной ряд

Неустойчивость предельных решений системы (1), по сути, делает некорректным применение классических численных методов на больших отрезках времени. Заметим, что на таких отрезках и строятся предельные множества динамических систем. Можно перейти к высокоточным вычислениям, но тогда придется брать достаточно малые шаги интегрирования. Решением проблемы является применение модификации метода степенных рядов с рекуррентным вычислением коэффициентов разложения.

Представим решение системы (1) в виде ряда

$$X(t) = \sum_{i=0}^{\infty} \Upsilon_i t^i, \tag{3}$$

где вектор Υ_0 определяется заданным начальным условием, остальные векторы-коэффициенты разложения (3) вычисляются по следующим рекуррентным формулам [4]:

$$\begin{gathered}
\Psi_i = [\psi_{i,1} \cdots \psi_{i,n}]^{\mathrm{T}}, \\
\psi_{i,p} = \sum_{j=0}^{i} \left\langle Q_p \Upsilon_i, \Upsilon_{i-j} \right\rangle, \quad p = 1,\ldots, n, \\
\Upsilon_1 = B_0 + B_1 \Upsilon_0 + \Psi_0, \\
\Upsilon_i = \frac{B_1 \Upsilon_{i-1} + \Psi_{i-1}}{i}, \quad i = 2,3,4,\ldots
\end{gathered} \tag{4}$$

Область сходимости ряда (3) ограничена для многих нелинейных систем вида (1), зависит от начального условия и может быть оценена по формулам [4]:

$$\begin{gathered}
h_1(\Upsilon_0) = \left\| \Upsilon_0 \right\|, \quad \mu = n \max_{p=1,\ldots,n} \left\| Q_p \right\|, \\
h_2(\Upsilon_0) = \begin{cases} \left\| B_0 \right\| + \left(\left\| B_1 \right\| + 2\mu \right) h_1(\Upsilon_0) + \mu \left(h_1(\Upsilon_0) \right)^2 & \text{для} \quad h_1 > 1, \\ \left\| B_0 \right\| + \left\| B_1 \right\| + \mu & \text{в противном случае,} \end{cases} \\
\tau(\Upsilon_0) = \frac{1}{h_2(\Upsilon_0) + \delta}, \quad t \in [-\tau; \tau],
\end{gathered} \tag{5}$$

где δ – любое положительное число (может быть выбрано сколь угодно малым).

Далее опишем алгоритм построения дуги траектории на заданном отрезке времени [0; T].

3. Алгоритм построения дуги траектории и программная реализация

В статье [5] численный метод, основанный на представлениях (3) – (5), называется FGBFI-методом (**F**irmly **G**rounded **B**ackward-**F**orward **I**ntegration – надежно обоснованное обратно-прямое интегрирование). Далее опишем алгоритм, положенный в его основу.

Пусть T – заданная длина отрезка интегрирования, S_a – шар, содержащий аттрактор системы (1). Зададим такое представление вещественного числа, чтобы машинный эпсилон

$$\varepsilon_m \ll \varepsilon_{pw},$$

где ε_{pw} – точность оценки общего члена ряда (3), т.е. суммирование по формуле (3) прекращается при таком значении $i = i^*$, когда

$$\left\| Y_{i^*} \right\| \cdot \left| \Delta t \right|^{i^*} < \varepsilon_{pw} , \qquad (6)$$

где Δt – шаг интегрирования. Отметим, что для сходимости ряда величину Δt нужно выбирать как

$$0 < \Delta t \le \tau(Y_0) \text{ (в прямом времени)}$$

или

$$-\tau(Y_0) \le \Delta t < 0 \text{ (в обратном времени).}$$

Т.к. область сходимости ряда (3) ограничена, мы не можем взять $\Delta t = T$ (как было указано выше, значение *T* велико). Поэтому будем сшивать дугу траектории на отрезке времени [0; *T*] дугами, на которых ряд (3) сходится.

Поскольку при компьютерном моделировании мы можем оперировать только дискретным набором векторов фазовых координат (вместо непрерывной функции *X*(*t*)), то по аналогии с методами Рунге-Кутты будем получать приближенные значения фазовых координат через шаг интегрирования, только вычисления вести на каждом шаге по формулам (3) – (6) при $t = \Delta t$.

Отметим, что здесь накопление ошибки на каждом шаге значительно меньше, чем для методов Рунге-Кутты фиксированного порядка, т.к. мы достаточно быстро можем досчитать нехватающие члены в сумме по формулам (3) и (4), уменьшив величину ε_{pw}. Из формул (5) видно, что при таком численном интегрировании шаг интегрирования будет переменным.

Пусть величина *way* отвечает за направление интегрирования: *way* = 1 – в прямом времени, *way* = –1 – в обратном времени.

Далее приведем алгоритм получения приближенных значений фазовых координат:

1. **Задать** вектор $X_0 \in S_a$ начального условия и значение *way*;
2. *t* := 0;
3. *ended* := `false`; /* заканчиваем ли алгоритм */
4. *l* := 1; /* номер отрезка сходимости ряда (3) */
5. **Вычислить** значение шага интегрирования $\Delta t = \tau(X_{l-1})$, используя формулы (5);
6. **Если** $\Delta t > T - t$, **то** $\Delta t := T - t$; *t* := *T*
   Иначе $t := t + \Delta t$;

7. $\mathrm{Y}_0 := X_{l-1}$;

8. $X_l := X_{l-1}$;

8. p := 1; /* произведение степеней t */

9. i := 0; /* номер текущего члена ряда */

10. $i := i + 1$; /* увеличить i на единицу */

11. $p := p \cdot way \cdot \Delta t$; /* вычислить текущую степень $t = \Delta t$ с учетом направления интегрирования – для нечетных степеней отрицательная величина p при $way = -1$ */

12. **Вычислить** по формулам (4) вектор Y_i;

13. $X_l := X_l + \mathrm{Y}_i \cdot p$; /* прибавить к X_l текущий член ряда */

14. $L := \|\mathrm{Y}_i\| \cdot |p|$; /* правая часть неравенства (6) */

15. **Если** $L > \varepsilon_{pw}$, **то перейти** к шагу 10;

16. **Вывести** значение текущего момента времени $way \cdot t$ и вектор X_l полученных фазовых координат;

17. **Если** $X_l \notin S_a$, **то** мы вышли за шар B_a; **вывести** "Уменьшите величину ε_{pw} и/или ε_m "; *ended* := `true`;

18. **Если** $t = T$, **то** *ended* := `true`;

19. **Если** *ended*, то **закончить алгоритм**;

17. $l := l + 1$;

18. **Перейти** к шагу 5.

Критерии проверки точности получаемого приближенного решения описаны в работах [4, 5]. Заметим, что обратный проход по времени используется в некоторых из них.

Для программной реализации приведенного выше алгоритма при $B_0 = \bar{0}$, а также исследования предельных точек динамических систем вида (1) на устойчивость по Пуассону и вычисления показателей Ляпунова модифицированным алгоритмом Бенеттина (см. статью [5]), был разработан комплекс программ на языке C++ под Linux. Получено свидетельство о государственной регистрации в Реестре программ для ЭВМ [6]. Исходные тексты доступны на GitHub по ссылке [7].

Для проведения матричных операций на языке C++ были использованы алгоритмы и шаблонные типы библиотеки uBLAS линейной алгебры собрания библиотек классов Boost. Отметим, что библиотека uBLAS дает большой выигрыш по времени.

Обычно в расчетах многие исследователи работают с субнормальными действительными числами одинарной или двойной точности, представленными в формате IEEE 754 [8]. Основным недостатком здесь является фиксированная точность представления действительных чисел, которая может не позволить нам численно построить приближения к неустойчивым решениям систем дифференциальных уравнений на больших отрезках времени. Поэтому для высокоточных вычислений используется библиотека MPFR C++ [9, 10, 11].

4. Заключение

В данной работе был описан высокоточный алгоритм построения приближений к неустойчивым решениям динамических систем вида (1), а также приведена ссылка на разработанный комплекс программ на языке C++. Для представления чисел произвольной точности использовался тип данных `mpreal` библиотеки MPFR C++ с перегруженными арифметическими операциями, а также дружественными математическими функциями.

Работа выполнена при финансовой поддержке РФФИ (грант 20-01-00347-а).

Список литературы

1. Lorenz, E. Deterministic nonperiodic flow / E. Lorenz // Journal of the Atmospheric Sciences. – 1963. – Vol. 20, Iss. 2. – pp. 130-141.
2. Hairer, E. Solving ordinary differential equations I: nonstiff problems / E. Hairer, S. P. Norsett, G. Wanner. – Berlin : Springer, 1993. – 528 p.
3. Butcher, J. C. Numerical methods for ordinary differential equations / J. C. Butcher. – Chichester : John Wiley & Sons, 2003. – 440 p.
4. Lozi, R. A new accurate numerical method of approximation of chaotic solutions of dynamical model equations with quadratic nonlinearities / R. Lozi, V. A. Pogonin, A. N. Pchelintsev // Chaos, Solitons & Fractals. – 2016. – Vol. 91. – pp. 108-114.
5. Pchelintsev, A. N. On the Poisson stability to study a fourth-order dynamical system with quadratic nonlinearities / A. N. Pchelintsev // Mathematics. – 2021. – Vol. 9, Iss. 17. – № 2057, 18 p.
6. Свидетельство о государственной регистрации программы для ЭВМ № 2021618416. Высокоточный метод решения систем обыкновенных дифференциальных уравнений с квадратичной правой частью / А. Н. Пчелинцев. – 23.07.2021 г.

7. The reliable calculations for the 4-th order system [Электронный ресурс]. – Режим доступа: https://github.com/alpchelintsev/4th_order_system

8. Overton, M. L. Numerical computing with IEEE floating point arithmetic / M. L. Overton. – Philadelphia : SIAM, 2001. – 106 p.

9. GNU MPFR library for multiple-precision floating point computations with correct rounding [Электронный ресурс]. – Электрон. журн. – Режим доступа: https://www.mpfr.org/

10. MPFR: A multiple-precision binary floating-point library with correct rounding / L. Fousse [and oth.] // ACM Transactions on Mathematical Software. – 2007. – Vol. 33, Iss. 2. – № 13, 15 p.

11. The high-performance C++ interface for MPFR library [Электронный ресурс]. – Режим доступа: https://github.com/advanpix/mpreal