

ДИФФЕРЕНЦИАЛЬНЫЕ УРАВНЕНИЯ
И

ПРОЦЕССЫ УПРАВЛЕНИЯ

№ 4, 2006

Электронный журнал,

рег. № П2375 от 07.03.97

ISSN 1817-2172

<http://www.neva.ru/journal>

<http://www.math.spbu.ru/user/diffjournal/>

e-mail: jodiff@mail.ru

Моделирование динамических систем

Применение методов интервальной арифметики к задаче построения символического образа

Н. Б. Ампилова, С. В. Терентьев

Россия, 198504, Университетский пр. д.28

Санкт-Петербургский Государственный Университет

Математико-механический факультет

e-mail: nataly@is1483.spb.edu, waterq@yandex.ru

Аннотация

При численном исследовании и моделировании динамических систем используется разбиение фазового пространства на ячейки. Такую ячейку удобно интерпретировать как интервальный вектор в пространстве соответствующей размерности и использовать аппарат интервальных вычислений.

В работе описан алгоритм построения символического образа дискретной динамической системы с помощью методов интервальной арифметики. Приводится ряд примеров и сравнение с методами, реализованными в обычной арифметике.

1 Введение

Одним из хорошо известных методов исследования динамических систем является построение по исходной системе ее аппроксимации — символического образа. По выбранному покрытию фазового пространства строится ориентированный граф, вершины которого соответствуют ячейкам покрытия, а дуги отвечают непустым пересечениям ячеек с их образами

при действии системы. Символический образ зависит от покрытия. Последовательное подразбиение ячеек покрытия позволяет получить более точное приближение к системе.

Такая конструкция позволяет применять алгоритмы на графах для получения многих важных характеристик динамической системы: инвариантных множеств и их изолирующих окрестностей, спектра Морса, энтропии.

Реализации алгоритмов построения символического образа основываются на так называемых *set-oriented* методах, в которых выбирается конечное покрытие исследуемой области фазового пространства и строятся приближения образов элементов (ячеек) покрытия. Любой подобный метод связан с построением некоторого множества в фазовом пространстве, которое наилучшим образом аппроксимировало бы образ ячейки. [4], [2]).

В работе [4] выбирается заданное число точек в каждой ячейке, их образы составляют образ ячейки. Основными стратегиями выбора являются однородное распределение точек внутри ячейки и выбор точек в окрестности границы.

В работе [2] рассматриваются линейный и точечный методы. Точечный метод выбирает равномерно разбросанные точки и в качестве образа ячейки строит объединение ячеек, содержащих образы точек. Линейный метод в качестве образа рассматривает наименьший n -мерный параллелепипед, ориентированный по осям координат, содержащий образы вершин ячейки.

Нам кажется естественным рассматривать ячейку фазового пространства как интервальный вектор в пространстве соответствующей размерности и использовать аппарат интервальной арифметики для реализации алгоритма построения образа ячейки. В отличие от обычной арифметики такой подход позволяет частично снять проблему с ошибками округления, поскольку мы всегда работаем не с числом, а с его окрестностью. При использовании интервальной арифметики может возникнуть так называемая проблема завышения результата, связанная с особенностью вычисления арифметических выражений: вхождения в формулу одной переменной трактуются как разные переменные. Специальная форма записи арифметических выражений помогает обойти это затруднение. Определенные трудности могут возникнуть при реализации интервальной арифметики в машинной. Здесь нужно вводить некоторые условия на характер машинного представления чисел. Как показано в [6], если интервальная функция удовлетворяет условию Липшица, то результат

вычислений в округленной интервальной арифметике стремится к результату вычислений в точной интервальной арифметике, когда машинная точность стремится к бесконечности. Иными словами, при подходящем выборе машинного представления результаты вычислений в арифметике машины имеют погрешность, зависящую от погрешности представления.

Разработан и реализован алгоритм построения символического образа с использованием библиотеки интервальных вычислений. Для интервальных функций вычисляются оценки констант Липшица. Приведены примеры, а также сравнительные характеристики работы алгоритма и алгоритмов, описанных в работе [2].

2 Интервальная арифметика

Мы приведем основные сведения из интервальной арифметики, которые потребуются в дальнейшем. Подробное изложение можно найти в [6].

Определение 1 *Интервалом называется множество вида :*

$$x = [\underline{x}, \bar{x}] = \{\tilde{x} \in R \mid \underline{x} \leq \tilde{x} \leq \bar{x}\}.$$

Радиус интервала x определяется следующим образом:

$$rad(x) = \frac{(\bar{x} - \underline{x})}{2}.$$

Пространство всех интервалов над R обозначается IR . На множестве IR вводится отношение частичного порядка. Пусть $x, y \in IR$, тогда

$$x \leq y \Leftrightarrow \bar{x} \leq \underline{y}.$$

Определение 2 *Пусть $x_1, \dots, x_m \in IR$. Тогда интервальным вектором называется множество*

$$\check{x} = (x_1, \dots, x_m).$$

Радиус интервального вектора определяется как

$$rad(\check{x}) = (rad(x_1), \dots, rad(x_m)), \quad x_1, \dots, x_m \in IR.$$

На пространстве всех интервальных векторов (IR^m) вводится отношение частичного порядка. Пусть $\check{x}, \check{y} \in IR^m$, тогда

$$\check{x} \leq \check{y} \Leftrightarrow \bar{\check{x}} \leq \underline{\check{y}},$$

где

$$\underline{\check{x}} = \sup(\underline{\bar{x}}_1, \dots, \underline{\bar{x}}_m)$$

$$\underline{\check{y}} = \inf(\underline{y}_1, \dots, \underline{y}_m).$$

Нетрудно заметить, что интервальный вектор имеет довольно естественное геометрическое представление: прямое произведение составляющих его интервалов.

Определение 3 Пусть $a, b \subset IR$. Тогда функция

$$q(a, b) = \sup \{ |\underline{a} - \underline{b}|, |\bar{a} - \bar{b}| \}$$

задает расстояние между интервалами a и b .

Легко проверить, что так введенное расстояние является метрикой на IR . В пространстве IR^m функция q определяется покомпонентно на интервалах. Заметим, что метрика q , вводимая на множестве интервалов, определяется и в том случае, когда интервалы пересекаются или содержатся один в другом, и в этом смысле отличается от расстояния между множествами. В случае непересекающихся интервалов она задает расстояние между множествами.

Вводится операция \square — построение оболочки множества из $R(R^m)$. Пусть $a \subset R$ — непустое ограниченное множество. Тогда

$$\square a = [\inf(a) , \sup(a)].$$

Иначе говоря, " \square " является операцией построения минимального интервала, содержащего множество a .

Пусть $a \subset R^m$ — непустое ограниченное множество. Тогда

$$\square a = [\inf(a_1) , \sup(a_1)] \times \dots \times [\inf(a_m) , \sup(a_m)],$$

где a_i — проекция множества a на i -ую ось, $i = 1, \dots, m$.

Обозначим через $\Omega = \{ +, -, *, /, ** \}$ множество элементарных операций в интервальной арифметике.

Для $x, y \in IR$ определим

$$x \circ y = \square \{ \tilde{x} \circ \tilde{y} \mid \tilde{x} \in x, \tilde{y} \in y \}, \circ \in \Omega.$$

Для $\check{x}, \check{y} \in IR^m$ определим

$$\check{x} \circ \check{y} = \square \{ [x_1 \circ y_1] \times \dots \times [x_m \circ y_m] \mid x_i \in \check{x}, y_i \in \check{y} \}, i \in \{1, \dots, m\}.$$

Вводится некоторое множество функций (Φ), называемых элементарными, которые являются вещественными функциями, непрерывными на каждом замкнутом интервале задания. Как правило, первоначально $\Phi = \{ \text{sqr}, \text{sqrt}, \text{ln}, \text{exp}, \text{sin}, \text{cos} \}$ и может быть расширено в процессе работы.

Для построения функций на интервалах (и интервальных векторах) — интервальных функций — вещественная прямая и множество интервалов расширяются с помощью символа ω (неопределенность):

$$R^* = R \cup \{\omega\}; IR^* = IR \cup \{\omega\}.$$

При этом предполагается, что ω больше любого вещественного числа.

Любая функция $\varphi \in \Phi$ расширяется на интервальный аргумент следующим образом:

$$\check{\varphi}(x) = \square \{ \varphi(\tilde{x}) \mid \tilde{x} \in x \}, x \in IR.$$

При выполнении некоторых условий монотонности интервальные функции есть расширение вещественных. Достаточно большой класс таких функций вводится с помощью арифметических выражений. Арифметическое выражение от формальных переменных (ξ_1, \dots, ξ_m) вводится как множество, содержащее R , множество переменных ξ_1, \dots, ξ_m , а также комбинации и композиции функций из Φ и элементарных операций из Ω . Для арифметического выражения $f = f(\xi_1, \dots, \xi_m)$ значение f в точке $x = (x_1, \dots, x_m) \in IR^m$ получается подстановкой интервалов x_l для соответствующих формальных параметров ξ_l , для $l = 1, \dots, m$.

3 Вычисление арифметических выражений в компьютерной арифметике

При реализации операций интервальной арифметики в арифметике машины нужно учитывать, что последняя представляет только конечное множество чисел. Обозначим его $H \subseteq R$ и введем операцию оптимального внешнего округления \diamond .

Определение 4 Пусть $x \in IR$, тогда $\diamond x$ есть минимальный интервал с координатами из H , содержащий x .

Определение 5 Пусть f — интервальная функция, заданная арифметическим выражением. Определим оператор " \diamond " следующим образом:

- $f = const$, тогда $\diamond f = const$;
- $f = \text{переменная}$, тогда $\diamond f = \text{переменная}$;
- $f = g \circ h$, тогда $\diamond f = \diamond(\diamond g \circ \diamond h)$, где $\circ \in \Omega$;
- $f = \varphi(g)$, тогда $\diamond f = \diamond\varphi(\diamond g)$, где $\varphi \in \Phi$.

Интервальные функции f и $\diamond f$ обладают следующим свойством:

$$\{f(\tilde{x}) \mid \tilde{x} \in x\} \subseteq f(x) \subseteq \diamond f(x)$$

для x из области определения f .

Для минимизации погрешности вычислений делаются определенные предположения о характере представления чисел в машинной арифметике. В частности, в работе [6] показано, что при представлении чисел в формате с плавающей точкой, результат полученный с использованием операции \diamond , может быть оценен через величины, связанные с этим представлением. Иными словами, результат операции округления остается в множестве H .

Вычисления функций в интервальной арифметике с учетом машинного представления опираются на следующий результат [6].

Теорема 1 При условии, что функция f задана арифметическим выражением от t переменных и удовлетворяет условию Липшица в $x \in IR^m$, найдутся такие положительные числа c_f и ε_f , что для любой округленной интервальной арифметики с относительной погрешностью $\varepsilon \leq \varepsilon_f$ функция $f^\diamond(x)$ определена и удовлетворяет неравенству

$$q(f^\diamond(x), f(x)) \leq c_f \varepsilon.$$

4 Символический образ динамической системы

Пусть M — замкнутое и ограниченное подмножество в R^q . Рассмотрим дискретную динамическую систему, порожденную гомеоморфизмом $f : M \rightarrow M$. Пусть $C = \{M_1, \dots, M_n\}$ — покрытие компакта M замкнутыми множествами. Множества $\{M_1, \dots, M_n\}$ назовем ячейками. По множеству

M и гомеоморфизму f построим ориентированный граф G следующим образом: ячейке M_i сопоставим i -ую вершину графа. Между вершинами i и j есть ориентированное ребро, если $f(M_i) \cap M_j \neq \emptyset$. Такой ориентированный граф называется символическим образом отображения f относительно S .

Для каждого i определим подпокрытие $C_i = \{M_j | M_j \cap f(M_i) \neq \emptyset\}$, множество $c_i = \{j | M_j \cap f(M_i) \neq \emptyset\}$ и множество $R_i = \bigcup_{j \in c_i} M_j$.

Обозначим за $diam(M_i) := \max \{\rho(x, y) | x, y \in M_i\}$ диаметр ячейки M_i . Наибольший из диаметров ячеек, обозначаемый d , называется диаметром символического образа.

Из построения d следует, что справедливы следующие включения :

$$f(M_i) \subseteq R_i \subseteq \{x | \rho(x, y) \leq d, y \in f(M_i)\} \quad (1)$$

Если ячейка M_k не принадлежит C_i , то определим расстояние между образом $f(M_i)$ и ячейкой M_k :

$$r_{ik} = \rho(f(M_i), M_k) = \min \{\rho(x, y), x \in f(M_i), y \in M_k\}$$

и положим

$$r = \min_{i,k} r_{ik}.$$

Число r называется нижней гранью символического образа. Диаметр и нижняя грань являются основными параметрами символического образа и зависят от выбора покрытия.

Справедливо следующее

Утверждение 1 [1]

1. $r \leq d$.
2. $\{x | \rho(x, f(M_i)) < r\} \subseteq R_i$.
3. Если $x \in M_j$ и $\rho(x, f(M_i)) < r$, то ячейка M_j принадлежит подпокрытию C_i .

Учитывая геометрический смысл интервальных векторов, удобно интерпретировать ячейки фазового пространства как интервальные вектора соответствующей размерности. Обозначим за \check{M}_i интервальный вектор, соответствующий ячейке M_i .

Тогда $C_i = \{\check{M}_j | \check{M}_j \cap f(\check{M}_i) \neq \emptyset\}$, $c_i = \{j | \check{M}_j \cap f(\check{M}_i) \neq \emptyset\}$ и $\check{R}_i = \bigcup_{j \in c_i} \check{M}_j$.

Диаметром ячейки назовем $diam(\check{M}_i) = \max_{j \in [1, m]} \{2rad(\check{M}_i)\}_j$ и обозначим через d наибольший из таких диаметров. Образ интервального вектора есть интервальный вектор, поэтому мы всегда работаем с n -мерными параллелепипедами в фазовом пространстве размерности n . Расстояние между ячейками естественно определять в обычном смысле расстояния между множествами. Таким образом, смысл величины r сохраняется.

5 Алгоритм построения символического образа с помощью интервальной арифметики

Основной задачей при построении символического образа является построение образа ячейки при действии исходной системы. Эта процедура реализуется с использованием библиотеки интервальных функций.

Символический образ дает все более точное приближение поведения системы при последовательном измельчении разбиения. Поэтому алгоритм подразумевает построение последовательности символических образов. Отметим, что в предлагаемом нами алгоритме мы фактически не строим граф, а работаем с множеством, содержащим ячейки, участвующие в формировании символического образа.

Пусть $f : \check{M} \rightarrow \check{M}$ — гомеоморфизм на компакте $\check{M} \subseteq IR^q$. Пусть $C_0 = \{\check{M}_1, \dots, \check{M}_n\}$ — начальное покрытие компакта \check{M} ячейками $\check{M}_i \subseteq IR^q; i = 1, \dots, n$. Для каждой ячейки \check{M}_i вычислим множества \check{R}_i . Построим множество-представление графа G , соответствующее символическому образу G отображения f относительно покрытия C_0 , а именно $G_{C_0} = \bigcup_{i \in [1, n]} \check{R}_i$. Это множество содержит все ячейки покрытия, которые участвуют в формировании графа G .

На следующем шаге разбиение применяется к элементам множества G_{C_0} , в результате повторения описанной процедуры получается множество-представление G_{C_1} для графа G_1 . На k -ом шаге мы рассматриваем разбиение вида

$$C_k = \{\check{M}_{j_1}, \dots, \check{M}_{j_m} \mid \check{M}_{j_i} \subset G_{k-1}\}$$

и строим соответствующее множество-представление G_{C_k} .

5.1 Представление ячейки.

В реализации алгоритма рассматриваются ячейки одного размера и целочисленная система координат, единица длины в которой есть размер ячейки. Это позволяет представить ячейку точкой ее левого верхнего угла единственным образом. Такое представление ячейки позволяет уменьшить объем памяти, требуемой для хранения данных.

5.2 Временная сложность алгоритма

При работе алгоритма основными операциями являются вычисление множеств \check{R}_i , добавление их в множество G_{C_k} и вычисление значения функции на интервальном векторе. Операция добавления требует $O(m)$ шагов, где m — количество элементов в множестве G_{C_k} . Вычисление \check{R}_i сводится к вычислению функции $f(\check{M}_i)$ и нахождению пересечения полученного образа с C_0 . В свою очередь вычисление функции заключается в вычислении выражения по дереву разбора этого выражения. Так как в реализации алгоритма используется целочисленная система координат, то операция пересечения сводится к сравнению целых чисел. Поэтому вычисление \check{R}_i происходит за $O(m_1 + m_2)$, где $O(m_1)$ — время вычисления выражения, $O(m_2)$ — время определения пересечения. Эти операции выполняются для каждой из n ячеек покрытия. В итоге время работы алгоритма оценивается как $O((m_1 + m_2)n) = O(n)$.

5.3 Сравнение алгоритма с точечным и линейным методами

Наиболее часто используемые методы построения образа ячейки ([2], [4]) в качестве аппроксимации образа получают некоторое множество. Сложность алгоритмов имеет порядок n или $n \log n$, где n — число элементов покрытия. В интервальном методе образом ячейки является интервальный вектор, который является результатом вычисления значения функции системы на ячейке и применения к ней операции " \square ". Эта операция строит наименьшее множество, содержащее результат.

Отметим, что в отличие от указанных работ мы не используем традиционную списочную структуру для хранения графа, а реализуем его представление с помощью задания ячейки в целочисленной системе координат. Как уже было отмечено, интервальный метод позволяет получить строгие оценки ошибок округления для элементарных операций.

Следует отметить, что современные реализации интервальной арифметики с оптимальным округлением при представлении чисел в формате с плавающей точкой поддерживают полубесконечные и бесконечные интервалы. Кроме того, применение этого метода позволяет вычислять значение константы Липшица для функции, задающей систему, по рекуррентной формуле параллельно с основными вычислениями.

В таблице 1 приведены результаты сравнения работы алгоритма с точечным и линейным методами [2]. Все вычисления использовали следующую аппаратную и программную конфигурацию:

- CPU: Intel Pentium 4 3Ghz
- Memory: 1 Gb
- OS: Microsoft Windows XP SP2.

В следующем далее тексте k_1 и k_2 обозначают ранг дробления ячеек по размерностям x и y соответственно, k — число последовательных итераций.

6 Примеры

6.1 Отображение Хенона

$$x_1 = 1 - 1.4x^2 + y, \quad y_1 = 0.3x.$$

У этой системы существует инвариантное множество в области $[-1.5, 1.5] \times [-\frac{10}{3}, \frac{10}{3}]$.

$$\check{M} = [-1.5, 1.5] \times [-10, 10],$$

$$k_1 = 1000, k_2 = 1000, k = 5.$$

Время работы — 40 сек. Инвариантное множество показано на рис.1.

6.2 Отображение Икеда [5]

$$x_1 = 2 - 0.9(x \cos \tau(x, y) - y \sin \tau(x, y)),$$

$$y_1 = 0.9(y \cos \tau(x, y) + x \sin \tau(x, y)),$$

где

$$\tau(x, y) = 0.4 - \frac{6}{1 + x^2 + y^2}.$$

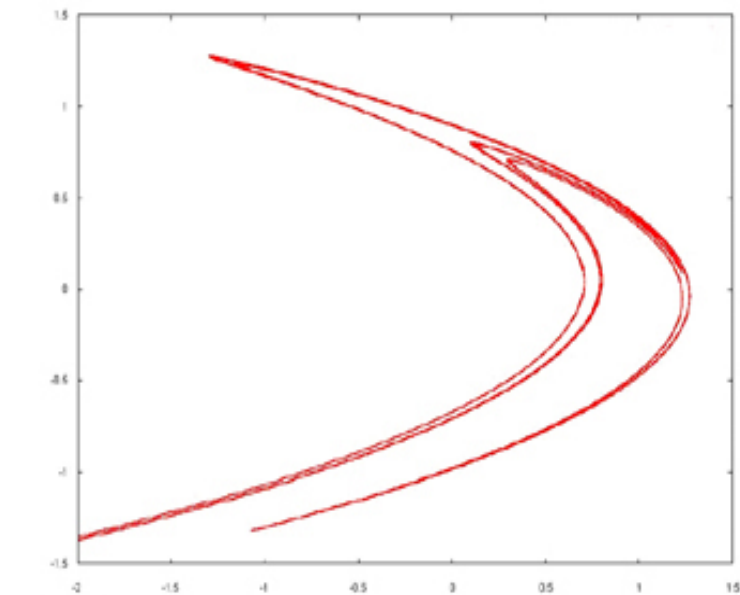


Рис. 1. Аттрактор Хенона

Вычисления проводились в области $\check{M} = [-10, 10] \times [-10, 10]$. Результат построения инвариантного множества показан на рис.2; $k_1 = k_2 = 1000, k = 86$. Время работы — 51 сек.

6.3 Отображение с задержкой

$$x_1 = y, \quad y_1 = ay(1 - x).$$

Система обладает двумя неподвижными точками $O_1 = (0, 0)$ и $O_2 = (1 - \frac{1}{a}, 1 - \frac{1}{a})$. Начало координат является седловой точкой при $a > 1$. Собственные числа в точке O_1 равны 0 и a . Точка O_2 является фокусом при $a > \frac{5}{4}$. Собственные числа в этой точке равны $\lambda_{1,2} = \frac{1}{2} (1 \pm \sqrt{5 - 4a})$. Фокус устойчив при $a > 2$, а при $a > 2$ фокус теряет устойчивость через бифуркацию Хопфа. Возникающая при этой бифуркации инвариантная кривая разрушается при $a = 2.27$ с появлением странного аттрактора [3].

Вычисления проводились в области $\check{M} = [-1.5, 1.5] \times [-10, 10]$ для $k_1 = 3200, k_2 = 3200$. На рисунках 3,4 и 5 показаны приближения к инвариантному множеству системы для $a = 2.27$ при значениях k равных 3, 15 и 22 соответственно. Последнее значение k дает довольно стабильное состояние, а именно, при дальнейшем увеличении числа итераций изображение не меняется. Время работы для этого значения составляет 22 мин 30 сек. Такое увеличение времени связано со сложной структурой

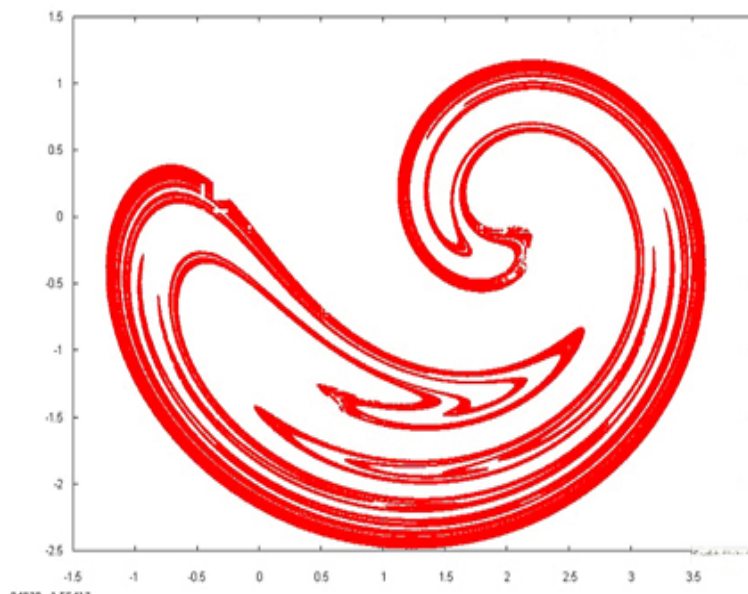


Рис. 2. Инвариантное множество отображения Икеда

инвариантного множества, а именно при $a = 2.27$ неустойчивая сепаратриса седловой неподвижной точки касается устойчивой. Моделирование этой ситуации требует большого числа итераций и достаточно мелкого разбиения. Алгоритм может быть улучшен за счет выбора формы области фазового пространства и оптимизации вычислений. Структура инвариантного множества вблизи седловой неподвижной точки показана на рис. 6.

Таблица 1

	Отображение Хенона	Отображение Икеда
метод	время работы	время работы
Линейный	24 047 ms	172 484 ms
Точечный	40 656 ms	112 359 ms
Интервальный	40 000 ms	51 000 ms

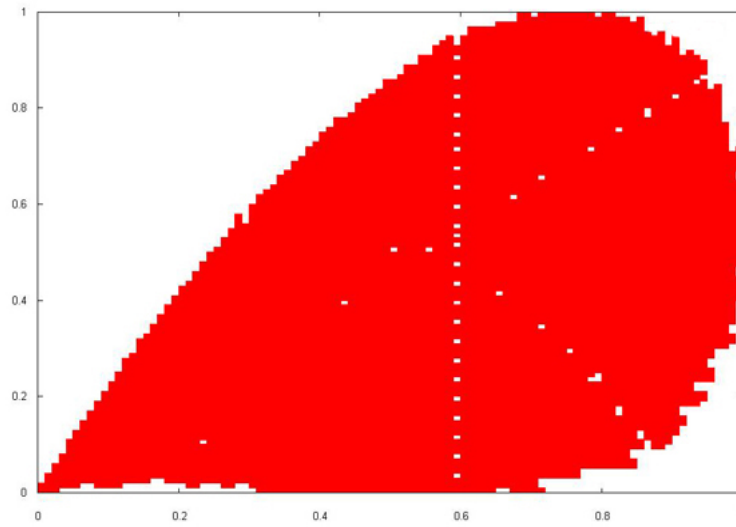


Рис. 3. $a = 2.27$. Инвариантное множество после 3 итераций

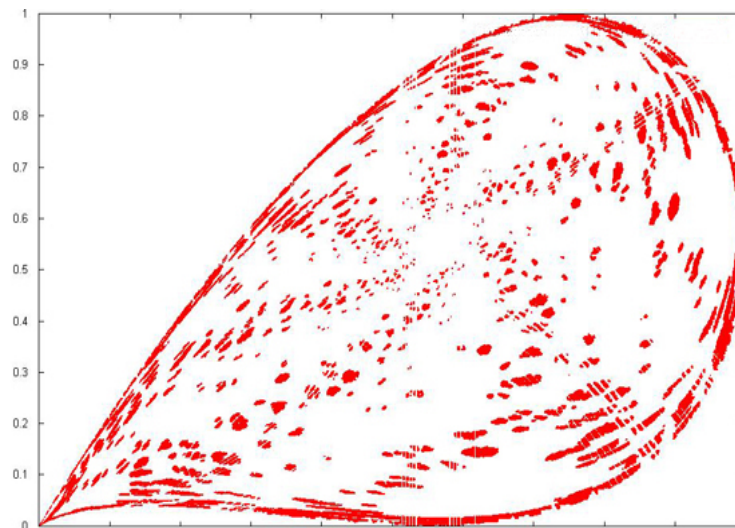


Рис. 4. $a = 2.27$. Инвариантное множество после 15 итераций

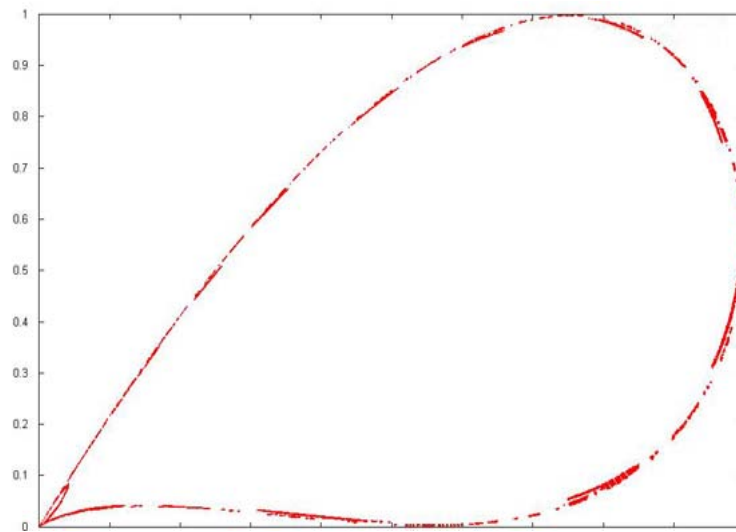


Рис. 5. $a = 2.27$. Инвариантное множество после 22 итераций

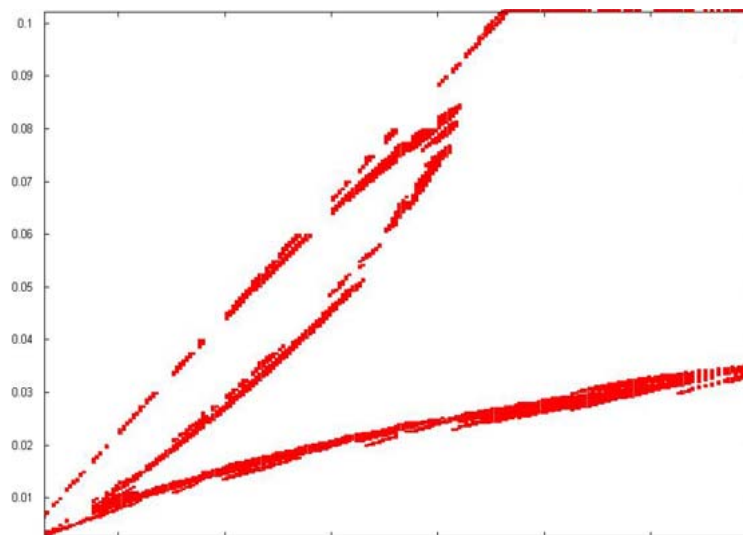


Рис. 6. $a = 2.27$. Инвариантное множество в окрестности седловой точки

Список литературы

- [1] Г.С. Осипенко, Н.Б. Ампилова. Введение в символический анализ динамических систем. изд. СПбГУ, 2005.
- [2] Петренко Е.И. Разработка и реализация алгоритмов построения символического образа. Электронный журнал "Дифференциальные уравнения и процессы управления", (<http://www.neva.ru/journal>), номер 3, 2006.
- [3] Aronson D.G., Chory M.A., Hall G.R. et.al Bifurcation from an invariant circle for two-parameter families of maps of the plane: A computer-assisted study. Commun.Math.Phys.83,3(1982), p.303-354.
- [4] Fundinger D. Implementation of the Symbolic Image. in: G. Osipenko. Dynamical System, Graphs, and Algorithms. Springer, 2007.
- [5] Ikeda K. Multiple-valued stationary state and its instability of the transmitted light by a ring cavity system, Opt. Commun. 30 (1979), p.257-261.
- [6] Neumaier A. Interval Methods for systems of equations, Cambridge University Press 1990.