

Алгоритм вычисления скользящего режима для системы с гладкой границей разрыва

В. В. КОРОБИЦЫН, Ю. В. ФРОЛОВА

Омский государственный университет, Россия

e-mail: korobits@rambler.ru

Описан численный алгоритм, предназначенный для решения систем обыкновенных дифференциальных уравнений, фазовое пространство которых состоит из двух областей непрерывности поля направлений системы с общим участком гладкой границы, где реализуется скользящий режим. Особенность алгоритма состоит в точном определении точки пересечения траектории решения с границей областей непрерывности. Для вычисления решения в скользящем режиме строятся две траектории по разные стороны от границы. Результаты вычислительного эксперимента показали сходимость предлагаемого алгоритма. Получена экспериментальная оценка вычислительных затрат алгоритма.

Ключевые слова: дифференциальные уравнения с разрывной правой частью, методы Рунге — Кутты, скользящий режим.

Введение

Обыкновенные дифференциальные уравнения (ОДУ) с разрывной правой частью активно используются для описания математических моделей разного рода процессов и систем. Исследованию динамических систем с разрывной правой частью, в том числе со скольжением, посвящены работы [1–4]. Такие системы получили различные названия: гибридные, релейные, кусочно-сшитые, системы Филиппова, динамические системы со скольжением и с разрывами, системы с клеточной структурой. Наиболее полное теоретическое описание систем ОДУ с разрывной правой частью дано в [5].

К настоящему времени имеется много работ, посвященных различным аспектам реализации алгоритмов численного решения ОДУ с разрывной правой частью [1–4]. Однако, проведя анализ существующих реализаций численных методов решения ОДУ [1–4], нельзя считать, что существуют готовые реализации методов решения систем Филиппова в общем виде.

В данной статье предлагается численный алгоритм, предназначенный для решения систем ОДУ с фазовым пространством, состоящим из двух областей непрерывности поля направлений системы с общим участком гладкой границы, на которой реализуется скользящий режим. В отличие от предыдущих работ авторов [1–4] здесь реализован алгоритм, имеющий три режима: непрерывный, пересечения границы и скольжения вдоль границы. Особенность алгоритма состоит в том, что для определения точки пересечения траектории решения с границей областей непрерывности правой части уравнения не используются значения функции правой части системы вне области непрерывности.

Для вычисления значения функции правой части системы на границе (в скользящем режиме) используются значения функций по разные стороны от границы. Таким образом, вдоль границы движутся две траектории численного решения, которые являются приближением точного решения с двух сторон. После выхода со скользящего режима две траектории сливаются в одну, и решение системы продолжается в непрерывном режиме.

В работе проведен вычислительный эксперимент с реализованным алгоритмом на примере, имеющем гладкую границу со скользящим режимом. Результаты показали сходимость алгоритма к точному решению. Получена экспериментальная оценка вычислительных затрат алгоритма.

1. Определение системы ОДУ на поверхности разрыва

Будем рассматривать автономную систему обыкновенных дифференциальных уравнений с кусочно-непрерывным полем направлений. Для таких систем характерно разбиение пространства состояний на отдельные области, в которых векторное поле является гладким. Границы между этими областями называют поверхностями разрыва. В общем виде система ОДУ записывается как

$$\frac{dx}{dt} = f(x), \quad x \in \mathbb{R}^n, \quad (1)$$

где векторное поле $f(x)$ может быть либо гладким, либо кусочно-гладким. Предположим, что пространство состояний состоит из двух областей G_i и G_j , разделенных поверхностью разрыва S_{ij} , которая определяется гладкой скалярной функцией $g_{ij}(x)$

$$S_{ij} = \{x \in \mathbb{R}^n : g_{ij}(x) = 0\},$$

а также

$$G_i = \{x \in \mathbb{R}^n : g_{ij}(x) > 0\}, \quad G_j = \{x \in \mathbb{R}^n : g_{ij}(x) < 0\}.$$

Тогда систему (??) можно переопределить как

$$\frac{dx}{dt} = \begin{cases} F_i(x), & x \in G_i, \\ F_j(x), & x \in G_j, \end{cases} \quad (2)$$

где F_i и F_j — достаточно гладкие функции. Будем считать, что эти функции определены в соответствующих областях G_i и G_j , а также на поверхности S_{ij} . При этом допускается, что функция F_i может быть не определена в области G_j и, наоборот, функция F_j может быть не определена в области G_i .

Если векторные поля F_i и F_j вблизи поверхности разрыва S_{ij} одновременно направлены к ней или от нее, то решение будет двигаться вдоль этой поверхности. Такое решение называют скользящим режимом. Открытое подмножество \hat{S}_{ij} поверхности S_{ij} , где векторные поля одновременно направлены к поверхности разрыва или от нее, обычно называют поверхностью скольжения. Кроме того, если справедливо неравенство

$$\mathcal{L}_{F_i - F_j}(g_{ij})(x) < 0, \quad x \in \hat{S}_{ij},$$

то поверхность разрыва \hat{S}_{ij} является устойчивой, но если

$$\mathcal{L}_{F_i - F_j}(g_{ij})(x) > 0, \quad x \in \hat{S}_{ij},$$

то \hat{S}_{ij} — неустойчива. Здесь через $\mathcal{L}_F(g)(x)$ обозначена производная Ли от функции $g(x)$ вдоль векторного поля $F(x)$

$$\mathcal{L}_F(g)(x) := \frac{dg}{dt}(x) = \frac{dg}{dx} \frac{dx}{dt}(x) = \left\langle \frac{dg}{dx}(x), F(x) \right\rangle.$$

На поверхности разрыва систему (??) можно доопределить как

$$\frac{dx}{dt} = F_{ij}(x), \quad x \in \hat{S}_{ij},$$

где

$$F_{ij}(x) = \frac{F_i(x) + F_j(x)}{2} + \frac{F_j(x) - F_i(x)}{2} \mu_{ij}(x), \quad (3)$$

а $-1 \leq \mu_{ij}(x) \leq 1$. Поскольку траектория решения должна двигаться по поверхности \hat{S}_{ij} , то векторное поле F_{ij} должно быть направлено по касательной к этой поверхности, т. е. $\mathcal{L}_{F_{ij}}(g_{ij})(x) = 0$. Отсюда

$$\mu_{ij}(x) = -\frac{\mathcal{L}_{F_i+F_j}(g_{ij})(x)}{\mathcal{L}_{F_j-F_i}(g_{ij})(x)}. \quad (4)$$

В результате получаем систему вида

$$\frac{dx}{dt} = \begin{cases} F_i(x), & x \in G_i, \\ F_{ij}(x), & x \in \hat{S}_{ij}, \\ F_j(x), & x \in G_j. \end{cases} \quad (5)$$

Такое доопределение системы (??) согласуется с подходом Филиппова, где система на поверхности разрыва заменяется дифференциальным включением с минимальным выпуклым множеством, содержащим F_i и F_j . Согласно теореме 2 из [?] (с. 85), учитывая гладкость функций F_{ij} и g_{ij} , решение системы (??) будет единственным справа, кроме точек на \hat{S}_{ij} , где оба вектора F_i и F_j ортогональны к \hat{S}_{ij} . Если оба вектора направлены к поверхности \hat{S}_{ij} , то эти точки являются стационарными точками системы, и решение будет единственным. Если же оба вектора направлены от поверхности разрыва, то поверхность \hat{S}_{ij} является неустойчивой. Решение, начинающееся с неустойчивой поверхности, не обладает единственностью. Однако если решение не начинается с точки в этой области, то и траектория решения не попадет на нее. Далее решения, начинающиеся с точки на неустойчивой границе, рассматриваться не будут, поэтому все исследуемые решения системы (??) будут обладать единственностью справа.

2. Точное определение точки входа траектории решения на поверхность разрыва

Для точного вычисления решения со скольльзящим режимом необходимо точно определить точку входа траектории решения на поверхность разрыва. Но мы будем искать не одну точку, а две, которые находятся достаточно близко от поверхности, но по разные от нее стороны. Обе точки должны быть близки к точному положению траектории решения на поверхности разрыва.

Выше было сделано предположение о том, что функция $F_i(x)$ может быть не определена в области G_j , а функция $F_j(x)$ — не определена в области G_i . Но для вычисления функции F_{ij} требуется определить значения функций F_i и F_j для одной и той же точки x , где $x \in G_i$ или $x \in G_j$.

Для точки $x \in G_i$ найдем близкую к ней точку $x^* \in G_j$ такую, что $|x - x^*| < \varepsilon|x|$, где ε — заданный предел локальной относительной погрешности решения. Такую точку можно найти, когда расстояние от точки x до поверхности S_{ij} достаточно мало (меньше $\frac{\varepsilon}{2}|x|$). Тогда можно принять точку x^* за представление точки x в области G_j и определить значения функций $F_j(x) := F_j(x^*)$ и, наоборот, $F_i(x^*) := F_i(x)$.

Для точек $x \in G_i$, лежащих достаточно далеко от границы, решение системы (??) будет определяться стандартной схемой численного решения ОДУ с непрерывной правой частью. Однако при приближении к поверхности разрыва для осуществления очередного шага интегрирования может потребоваться значение функции правой части из области G_j . Но поскольку функция F_i в области G_j не определена, то возникает вопрос — как найти длину шага интегрирования и точку пересечения траектории с поверхностью разрыва? Предлагаем сделать это в два этапа: сначала определить длину шага интегрирования такую, что очередная точка будет найдена в области G_i , но достаточно близко к границе, а затем, используя экстраполяцию кривой решения, найти две точки $x_b \in G_i$ и $x_b^* \in G_j$, лежащие на продолжении траектории решения, но по разные стороны от поверхности разрыва.

Для первого этапа выберем шаг

$$h_1 = -K \frac{g_{ij}(x)}{\mathcal{L}_{F_i}(g_{ij})(x)}, \quad (6)$$

где $K > 0$ — константа, регулирующая приближение к границе (чтобы найденное значение решения осталось в области G_i , можно задать $K = 0.9$). Сделаем шаг интегрирования h из точки x_k в точку x_{k+1} .

Для экстраполяции кривой решения до границы воспользуемся интерполяционной формулой Ньютона для интерполирования назад, опираясь на три точки $x_k \approx x(t_k - h)$, $x_{k+1/2} \approx x(t_k - h/2)$ и $x_{k+1} \approx x(t_k)$. Точка $x_{k+1/2}$ будет получена при нахождении точки x_{k+1} , если используется схема Рундсона для оценки погрешности решения. В точках $x_k, x_{k+1/2}, x_{k+1}$ можно вычислить значения функции F_i : $f_k, f_{k+1/2}, f_{k+1}$, причем первые два значения были вычислены при нахождении точки x_{k+1} . Обозначим также $d_1 = (x_{k+1} - x_{k+1/2})/h$, $d_2 = (x_{k+1/2} - x_k)/h$. Тогда интерполяционный многочлен будет иметь вид

$$x(t_k + \theta h) \approx N(\theta) = x_{k+1} + \theta h \left[f_{k+1} + \theta \left(d_1 - f_{k+1} + (\theta - 1) \left(f_{k+1/2} - 2d_1 + f_{k+1} + \frac{\theta - 1}{4} \left(d_2 - 4f_{k+1/2} + 5d_1 - 2f_{k+1} + \frac{\theta - 2}{4} \cdot (f_k - 3d_2 + 4f_{k+1/2} - 3d_1 + f_{k+1}) \right) \right) \right) \right]. \quad (7)$$

Построенный многочлен имеет четвертый порядок точности.

Чтобы найти точки x_b и x_b^* , необходимо определить соответствующие θ_l и θ_r такие, что $x_b = N(\theta_l)$ и $x_b^* = N(\theta_r)$. Для поиска этих значений воспользуемся итерационным методом Ньютона для решения алгебраического уравнения

$$g_{ij}(x) = 0, \text{ где } x = N(\theta).$$

Итерационная процедура имеет вид

$$\theta_{s+1} = \theta_s - \mathcal{C} \cdot \frac{g_{ij}(x_s)}{h \cdot \mathcal{L}_{F_i}(g_{ij})(x_s)}, \quad \theta_0 = 0.5, \quad s = 0, 1, \dots$$

Здесь в качестве производной от функции g_{ij} используется производная по направлению $\mathcal{L}_{F_i}(g_{ij})(x_s)$, умноженная на h (производная сложной функции $g'_{ij}(x) \cdot x'(\theta)$). Значение точки $x_s = x(t_k + \theta_s h)$ вычисляется по формуле (??). Коэффициент $\mathcal{C} = 1.1$ позволяет постоянно “перепрыгивать” через границу, обеспечивая приближение к ней с обеих сторон. Осуществляя итерационную процедуру, необходимо сохранять величину $\theta_l = \max_s \{\theta_s\}$, для которой $x(t_k + \theta_s h) \in G_i$, и $\theta_r = \min_s \{\theta_s\}$, для которой $x(t_k + \theta_s h) \in G_j$. Таким способом будем определять наиболее близкие точки по разные стороны от поверхности S_{ij} , лежащие на продолжении кривой решения. Если в ходе итерационной процедуры будет выполнено условие $|\theta_s| > 1$, то необходимо остановить процедуру по причине расходимости метода. Успешным завершением процедуры является выполнение условия

$$|\theta_l - \theta_r| \leq \left| \kappa \cdot \frac{\varepsilon}{\eta} \right|,$$

где κ — константа (экспериментально установлено значение $\kappa = 0.0005$, позволяющее найти точку пересечения с заданной точностью ε), значение η определяется по формуле

$$\eta = \frac{\tau}{1 - \tau}, \quad \text{где} \quad \tau = \frac{\Delta\theta_s}{\Delta\theta_{s-1}}, \quad \Delta\theta_s = \theta_s - \theta_{s-1}, \quad s = 2, 3, \dots$$

Вычисление значения константы κ требует дополнительного исследования, которое выходит за рамки данной статьи.

3. Условия продолжения решения на поверхности разрыва

Имея значения точек x_b и x_b^* , можно продолжить решение системы после достижения поверхности разрыва. При этом о дальнейшем поведении решения можно сделать вывод, анализируя значения вектор-функций $F_i(x_b)$ и $F_j(x_b^*)$ по отношению к поверхности S_{ij} . Если выполнено условие пересечения

$$\mathcal{L}_{F_i}(g_{ij})(x_b) \cdot \mathcal{L}_{F_j}(g_{ij})(x_b^*) > 0, \quad (8)$$

то траектория решения должна покинуть поверхность разрыва. При $\mathcal{L}_{F_i}(g_{ij})(x_b) < 0$ решение перейдет в G_i и продолжить его необходимо из точки x_b , при $\mathcal{L}_{F_i}(g_{ij})(x_b) > 0$ решение перейдет в G_j и продолжить его необходимо из точки x_b^* . Если эти векторы одновременно направлены к поверхности разрыва S_{ij} или от нее:

$$\mathcal{L}_{F_i}(g_{ij})(x_b) \cdot \mathcal{L}_{F_j}(g_{ij})(x_b^*) < 0, \quad (9)$$

то реализуется скользкий режим. Причем если первый множитель $\mathcal{L}_{F_i}(g_{ij})(x_b) > 0$ (векторы направлены к поверхности), то скользкий режим является устойчивым (поверхность — притягивающей), и наоборот, при $\mathcal{L}_{F_i}(g_{ij})(x_b) < 0$ (векторы направлены от поверхности) скользкий режим является неустойчивым (поверхность — отталкивающей). В последнем случае траектория решения может “соскользнуть” с поверхности

в любой момент, что не дает возможности говорить об единственности решения системы. Однако возможна ситуация, когда выполняется равенство

$$\mathcal{L}_{F_i}(g_{ij})(x_b) \cdot \mathcal{L}_{F_j}(g_{ij})(x_b^*) = 0. \quad (10)$$

В этом случае, если один из сомножителей не равен нулю, то на поверхности разрыва одно из векторных полей направлено ортогонально к поверхности, а проекция другого вектора определяет скорость скольжения вдоль поверхности. Вместе с тем, если оба сомножителя равны нулю, то рассматриваются условия более высокого порядка — вторые производные функции $g_{ij}(x)$ [?].

4. Две скользящие траектории вблизи поверхности разрыва

Найденные выше точки x_b и x_b^* представляют собой пару, которая служит приближением точки решения $x(t_{k+1})$ на поверхности разрыва S_{ij} , причем

$$t_{k+1} = t_k + \frac{\theta_l + \theta_r}{2} h.$$

Для продолжения решения рассмотрим случай скользящего режима.

Предлагаем находить не одну траекторию решения, а две — по обе стороны от поверхности разрыва. Одна траектория стартует из точки x_b , вторая — из точки x_b^* . Траектории будем искать как решение системы

$$\frac{dx}{dt} = F_{ij}(x; x^*), \quad \frac{dx^*}{dt} = F_{ji}(x^*; x), \quad x \in G_i, \quad x^* \in G_j, \quad t > t_{k+1} \quad (11)$$

с начальными данными

$$x(t_{k+1}) = x_b, \quad x^*(t_{k+1}) = x_b^*.$$

Функция F_{ij} определяется формулами (??), (??) с заменой аргумента x на x^* для функции F_j :

$$F_{ij}(x; x^*) = \frac{F_i(x) + F_j(x^*)}{2} + \frac{F_j(x^*) - F_i(x)}{2} \cdot \mu_{ij}(x; x^*),$$

$$\mu_{ij}(x; x^*) = - \frac{\left\langle \frac{dg_{ij}}{dx}(x), F_i(x) \right\rangle + \left\langle \frac{dg_{ij}}{dx}(x^*), F_j(x^*) \right\rangle}{\left\langle \frac{dg_{ij}}{dx}(x^*), F_j(x^*) \right\rangle - \left\langle \frac{dg_{ij}}{dx}(x), F_i(x) \right\rangle}, \quad (12)$$

причем $F_{ij}(x; x^*) = F_{ji}(x^*; x)$, а это означает, что скорости движения вдоль поверхности разрыва точек траекторий с обеих сторон будут одинаковы. При этом $x(t) \in G_i$, $x^*(t) \in G_j$ для любых $t \geq t_{k+1}$.

Движение траектории в скользящем режиме может прекратиться, и тогда траектория либо продолжит движение в непрерывном (не скользящем) режиме, либо достигнет устойчивой точки равновесия и останется в ней, либо продолжит движение в скользящем режиме, но по другой поверхности разрыва. Прекращение скользящего режима возможно при достижении траектории решения точки, в которой выполнимо одно из трех условий: 1 — векторы правых частей направлены по одну сторону от поверхности разрыва; 2 — точка является особой устойчивой; 3 — точка лежит на пересечении с другой поверхностью разрыва. В первом случае решение необходимо продолжить из

точки, лежащей по одну сторону от поверхности разрыва с вектором правой части, во втором — траектория решения останется в особой точке, в третьем — поведение решения будет зависеть от векторных полей по разные стороны от обеих поверхностей разрыва и в данной работе не рассматривается.

5. Описание алгоритма

Предлагаем алгоритм вычисления численного решения кусочно-сшитой системы (Piecewise Sewing System) обыкновенных дифференциальных уравнений (??) с точным вычислением скользящего режима.

Алгоритм. Обеспечивается вычисление решения кусочно-сшитой системы обыкновенных дифференциальных уравнений на интервале времени $[a, b]$ для начального значения x_0 . Решение сохраняется в массивы T и X . Используется схема Рунге — Кутты для интегрирования уравнения в области непрерывности. Для поиска решения на границе применяется метод экстраполяции полиномом Ньютона (??). Для нахождения решения в скользящем режиме используется метод Рунге — Кутты для системы уравнений (??), (??).

Шаг 0 (*инициализация*). Установим начальные значения $x \leftarrow x_0$, $t \leftarrow a$, $i \leftarrow 0$. Сохраним в массив $T[i] \leftarrow t$, $X[i] \leftarrow x$, $i \leftarrow i + 1$. Выберем начальный шаг интегрирования h . Установим режим $mode \leftarrow cont$.

Шаг 1 (*выбор режима вычислений*). Если $mode = cont$, то перейдем на шаг 2, если $mode = intrs$ — на шаг 3, если $mode = slide$ — на шаг 5.

Шаг 2 (*непрерывный режим*). Сделаем шаг интегрирования h методом Рунге — Кутты. Если произошло пересечение поверхности разрыва, то выберем длину шага h_1 согласно формуле (??), немного не доходя до пересечения границы, выполним шаг $h \leftarrow h_1$ и переключим режим $mode \leftarrow intrs$. Перейдем на шаг 6.

Шаг 3 (*режим пересечения*). Найдем значение двух точек x_b и x_b^* , лежащих на продолжении траектории решения: до и после пересечения границы, расстояние между точками должно быть меньше заданного предела погрешности ε (см. разд. 2). Если алгоритм не сходится, то уменьшим шаг $h \leftarrow h/2$, переключим режим $mode \leftarrow cont$ и перейдем на шаг 1, иначе — на шаг 4.

Шаг 4 (*скольжение или продолжение?*). Если условие скольжения (??) выполнено, то сохраним среднюю точку и обе точки используем для скользящего режима $mode \leftarrow slide$. Если условие скольжения не выполнено, то сохраним обе точки и продолжим решение в области выхода траектории в непрерывном режиме $mode \leftarrow cont$. Перейдем на шаг 6.

Шаг 5 (*режим скольжения*). Выполним шаг интегрирования методом Рунге — Кутты для скользящего режима. Если произошло пересечение границы области, то выберем длину шага h_2 до пересечения границы, выполним шаг $h \leftarrow h_2$ до границы и переключим режим $mode \leftarrow intrs$. Если скользящий режим завершился, то $mode \leftarrow cont$.

Шаг 6 (*сохранить точку решения и продолжить вычисления?*). Если погрешность найденной точки решения меньше заданной точности, то сохраним результат $t \leftarrow t + h$, $T[i] \leftarrow t$, $X[i] \leftarrow x$, $i \leftarrow i + 1$. Если $t \leq b$, то вернемся к шагу 1, иначе — закончим вычисления.

Решение системы (??) по представленному алгоритму определяется тремя режимами: непрерывный, пересечение и скольжение. Непрерывный режим реализуется одной из схем Рунге — Кутты. Например, можно использовать схемы Мерсона, Фельберга или Дормана — Принса [?], а также многошаговые методы, но в нашем вычислительном эксперименте была применена схема Рунге — Кутты — Фельберга.

Вычисление правых частей системы уравнений в произвольной точке x реализуется в два этапа: определение номера i области G_i , содержащей x , и вычисление соответствующей функции $F_i(x)$. Определение номера области сводится к вычислению логических функций, определяющих условия ограничения пространства области. Однако при вычислении решения вблизи границы по схеме Рунге — Кутты может возникнуть ситуация, когда для одного шага по схеме необходимо вычислить значения функции правой части в разных областях G_i и G_j . Это означает, что возможно пересечение границы областей траекторией. Поэтому необходимо более точно определить точку пересечения траектории решения с поверхностью разрыва, для чего реализован режим пересечения. Схема определения точек пересечения траектории решения с поверхностью разрыва приведена выше в разделе 2.

После нахождения точек пересечения необходимо определить дальнейшее поведение решения: продолжить решение в другой области или перейти в скользящий режим (условия определения поведения (??), (??)). Если выполнено условие пересечения, то обе точки x^* и x_b^* сохраняются в выходные массивы X и T , а решение продолжается из точки x_b^* в непрерывном режиме. Если же выполнено условие скольжения, то в выходные массивы сохраняется одна точка $t^s = (t^* + t_b^*)/2$, $x^s = (x^* + x_b^*)/2$. Из обеих точек x^* , x_b^* стартуют траектории решения в скользящем режиме, являющиеся решением системы (??).

Численное решение в скользящем режиме находится методом Рунге — Кутты для правой части (??). Причем при вычислении правой части системы (??) можно сократить количество выполняемых арифметических операций, учитывая условие $F_{ij}(x; x^*) = F_{ji}(x^*; x)$. Также отметим, что значение производной функции $g_{ij}(x)$ можно заменить выражением

$$\frac{dg_{ij}}{dx}(x) \approx \frac{dg_{ij}}{dx}(x^*) \approx \frac{g_{ij}(x^*) - g_{ij}(x)}{x^* - x}.$$

Завершение скользящего режима так или иначе связано с пересечением одной из траекторий или обеими траекториями одновременно некоторой поверхности разрыва. Если одна из траекторий перескочила на другую сторону поверхности разрыва (или обе поменялись сторонами), то либо траектория должна покинуть поверхность в силу завершения скользящего режима, либо шаг интегрирования необходимо уменьшить и продолжить скольжение. Если же одна из траекторий или обе пересекли другую поверхность разрыва, то это означает, что траектория достигла точки пересечения поверхностей разрыва и необходимо детально изучить дальнейшее направление движения траектории.

Если оценка погрешности найденной точки решения на данном шаге меньше заданной точности, то значение точки сохраняется в массивы X и T . Если же шаг оказался

неудачным, то итерация алгоритма повторяется снова, стартуя с предыдущей сохраненной точки. Оценка погрешности численного решения производится с помощью метода Рундсона. Дополнительно вычисляется решение с двумя шагами половинной длины. Эта схема используется как для непрерывного, так и для скользящего режима. Вложенные схемы оценки погрешности показали себя хуже метода Рундсона [?], поэтому в настоящей работе не использовались. В режиме пересечения погрешность оценивается условием завершения итерационной процедуры метода Ньютона.

При достижении конечной точки интервала времени $t = b$ алгоритм завершает свою работу.

6. Вычислительный эксперимент

Простой пример скользящего режима можно продемонстрировать на решении следующей системы. Состояние системы описывается вектором $(x_1, x_2) \in \mathbf{R}^2$. Система имеет две области G_1 и G_2 непрерывного решения, разделенные кривой разрыва $g(x_1, x_2) := x_2 = 0$, $G_1 = \{(x_1, x_2) : x_2 > 0\}$, $G_2 = \{(x_1, x_2) : x_2 < 0\}$. В области G_1 решение системы задается системой дифференциальных уравнений

$$\frac{dx_1}{dt} = -0.1(x_2 - 2), \quad \frac{dx_2}{dt} = 0.3(x_1 - 1), \quad (x_1, x_2) \in G_1, \quad (13)$$

в области G_2 —

$$\frac{dx_1}{dt} = -0.1(x_2 - 1), \quad \frac{dx_2}{dt} = 0.3(x_1 + 1), \quad (x_1, x_2) \in G_2. \quad (14)$$

Фазовый портрет системы (рис. ??, а) склеен из двух областей G_1 и G_2 , решение в каждой из которых представляет собой цикл вокруг особой точки (тип особой точки — центр), причем область G_1 содержит особую точку $(1; 2)$ для функции F_1 , а особая точка $(-1; 1)$ для функции F_2 в область G_2 не попадает. Траектории численного решения движутся против часовой стрелки. При склеивании областей образуется участок границы со скользящим режимом с направлением движения от точки $(-1; 0)$ до точки $(1; 0)$. В остальной части границы пересечение происходит на тех ее участках, где траектории

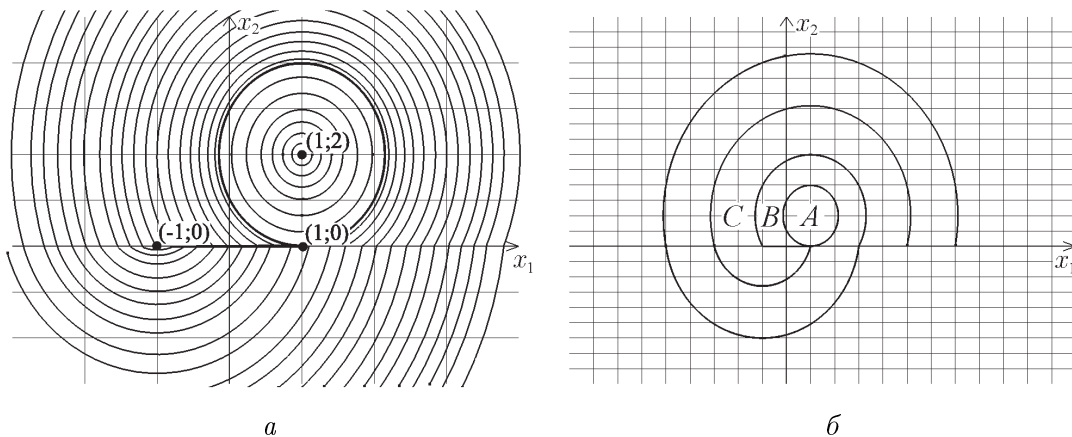


Рис. 1. Фазовый портрет системы (??), (??) (а); области фазовой плоскости (б)

непрерывны, но не являются гладкими, причем при $x_1 < -1$ траектории пересекают границу $x_2 = 0$ сверху вниз, а при $x_1 > 1$ — снизу вверх.

Условно все фазовое пространство системы можно разделить на три непересекающиеся области A, B, C (рис. ??, б), траектории в которых имеют разное поведение. Область A представляет собой эллипс с центром в точке $(1; 2)$, касающийся границы $x_2 = 0$. Внутри этой области траектории являются циклическими и образуют концентрические эллипсы. Область B — спиральная полоса, наворачивающаяся на область A . Траектории из этой области несколько раз пересекают границу областей G_1 и G_2 , затем входят в скользящий режим сверху и, сместившись до точки $(1; 0)$, входят в цикл, представляющий собой границу области A . Область C также является спиральной полосой, но в этом случае траектории входят в скользящий режим снизу и далее переходят на границу области A .

Для вычислительного эксперимента, исследующего поведение предложенного численного алгоритма, используем систему (??), (?). Нас будет интересовать поведение алгоритма в трех режимах: непрерывный, пересечение и скользящее. Все эти режимы реализуются в предложенной системе ОДУ.

В качестве тестовых выберем по одной траектории из каждой области A, B, C . Зададим начальные точки: $(2; 2)$ из A , $(0; -4)$ из B , $(0; 7)$ из C . Первая траектория находится в области непрерывности системы и требуется для изучения поведения алгоритма в условиях непрерывной задачи. Вторая (третья) траектория сначала проходит точку пересечения границы, затем входит в скользящий режим сверху (снизу), выходит со скользящего в точку $(1; 0)$ и переходит в циклическое решение на границе области A . Эти траектории позволяют изучить поведение алгоритма в трех ситуациях: пересечение границы разрыва, переход траектории в скользящий режим и выход из скользящего.

Точное решение системы (??), (??) в областях G_1 и G_2 легко вычислить аналитически. Решение в общем виде выглядит как

$$x_1(t) = C_1 \cos\left(\frac{\sqrt{3}}{10}(t - t_0)\right) + C_2 \sin\left(\frac{\sqrt{3}}{10}(t - t_0)\right) + \chi_1,$$

$$x_2(t) = C_1 \sqrt{3} \sin\left(\frac{\sqrt{3}}{10}(t - t_0)\right) - C_2 \sqrt{3} \cos\left(\frac{\sqrt{3}}{10}(t - t_0)\right) + \chi_2,$$

где константы C_1 и C_2 определяются исходя из начальных данных траектории решения $C_1 = x_1(t_0) - \chi_1$, $C_2 = -\frac{\sqrt{3}}{3}(x_2(t_0) - \chi_2)$, где χ_1, χ_2 — координаты особой точки в областях G_1, G_2 . Из найденного общего решения можно сконструировать решения для интересующих нас траекторий из областей A, B, C .

Для области A с заданной начальной точкой $x_1(0) = 2, x_2(0) = 2$ получаем решение

$$x_1(t) = \cos\left(\frac{\sqrt{3}}{10}t\right) + 1, \quad x_2(t) = \sqrt{3} \sin\left(\frac{\sqrt{3}}{10}t\right) + 2.$$

Это решение является циклическим с периодом $t_A = \frac{20\sqrt{3}\pi}{3}$.

Для области B с заданной начальной точкой $x_1(0) = 0$, $x_2(0) = -4$ получаем решение, склеенное из четырех траекторий (I, II, III, IV):

$$x_1^I(t) = \cos\left(\frac{\sqrt{3}}{10}t\right) + \frac{5\sqrt{3}}{3}\sin\left(\frac{\sqrt{3}}{10}t\right) - 1,$$

$$x_2^I(t) = \sqrt{3}\sin\left(\frac{\sqrt{3}}{10}t\right) - 5\cos\left(\frac{\sqrt{3}}{10}t\right) + 1.$$

Эта траектория достигает границы в точке $x_1^I(t_1) = 2$, $x_2^I(t_1) = 0$ в момент времени $t_1 = \frac{10\sqrt{3}\pi}{9}$. Далее решение продолжается в области G_1 траекторией

$$x_1^{II}(t) = \cos\left(\frac{\sqrt{3}}{10}(t - t_1)\right) + \frac{2\sqrt{3}}{3}\sin\left(\frac{\sqrt{3}}{10}(t - t_1)\right) + 1,$$

$$x_2^{II}(t) = \sqrt{3}\sin\left(\frac{\sqrt{3}}{10}(t - t_1)\right) - 2\cos\left(\frac{\sqrt{3}}{10}(t - t_1)\right) + 2,$$

достигающей границы в точке $x_1^{II}(t_2) = 0$, $x_2^{II}(t_2) = 0$ в момент времени $t_2 = t_1 + \frac{10\sqrt{3}}{3}\left(\pi + \arccos\left(-\frac{1}{7}\right)\right)$. Затем решение продолжается траекторией на границе в скользящем режиме и описывается функциями

$$x_1^{III}(t) = 0.15(t - t_2), \quad x_2^{III}(t) = 0.$$

Далее траектория достигает точки выхода из скользящего режима $x_1^{III}(t_3) = 1$, $x_2^{III}(t_3) = 0$ в момент времени $t_3 = t_2 + \frac{100}{15}$. Затем решение продолжается циклически

$$x_1^{IV}(t) = \frac{2\sqrt{3}}{3}\sin\left(\frac{\sqrt{3}}{10}(t - t_3)\right) + 1,$$

$$x_2^{IV}(t) = -2\cos\left(\frac{\sqrt{3}}{10}(t - t_3)\right) + 2,$$

и траектория вновь достигает точку $x_1^{IV}(t_4) = 1$, $x_2^{IV}(t_4) = 0$ в момент времени $t_4 = t_3 + \frac{20\sqrt{3}\pi}{3}$.

Аналогично приведенной конструкции траектории решения системы в области B можно получить решение в области C .

Вычислительный эксперимент проводился для оценки эффективности работы предложенного алгоритма. Фазовая плоскость системы была разделена на три области (см. рис. ??, б), в каждой из которых траектории имеют качественно одинаковое поведение. Поэтому было выбрано по одной тестовой траектории в каждой области, найдены для них аналитические решения (см. выше) и вычислены несколько численных решений с разной заданной точностью для каждой начальной точки. Отслеживание траектории

проведено на интервале времени от заданной начальной точки до завершения первого цикла на границе области A .

Траектория решения в области A (рис. ??, a) находится в области непрерывности векторного поля направления правой части системы. Эта траектория используется для проверки работы алгоритма в стандартной (непрерывной) ситуации с целью определения — не происходит ли увеличения времени счета по сравнению с таковым для традиционных алгоритмов и не нарушается ли сходимость метода на непрерывно-дифференцируемом решении. Как показывают результаты вычислений (рис. ??, b), алгоритм сохраняет сходимость к точному решению. На рисунке приведены графики изменения порядка абсолютной погрешности $\lg A_{err}$ численного решения при разной заданной локальной точности $Tol = 10^{-2}, 10^{-4}, 10^{-6}, 10^{-8}$. Видно, что погрешность получаемых решений не превышает заданной локальной точности.

Траектория решения в области B (рис. ??, a) стартует из точки $(0; -4)$, пересекает границу областей непрерывности в точке $(2; 0)$, где происходит смена направления векторного поля правой части системы, затем снова достигает границы в точке $(0; 0)$, где переходит в скользящий режим вдоль границы. В точке $(1; 0)$ траектория выходит со скольжения и продолжает движение по циклу на границе области A . Графики погрешности численного решения (рис. ??, b) показывают сходимость к точному решению при уменьшении значения заданной точности Tol . Однако в отличие от непрерывного режима (см. рис. ??, a) при переходе через границу при входе в скользящий режим и выходе из него возникает скачок погрешности, что, как следствие, ведет к увеличению количества вычислительных операций для нахождения решения с заданной точностью (см. ниже рис. ??, a). Тем не менее и в этом случае удается найти решение с необходимой точностью.

Траектория решения в области C (рис. ??, a) ведет себя аналогично траектории из области B . Отличие состоит лишь в том, что на участок скользящего режима траектория приходит снизу (из области G_2), а не сверху (из области G_1), как в предыдущем

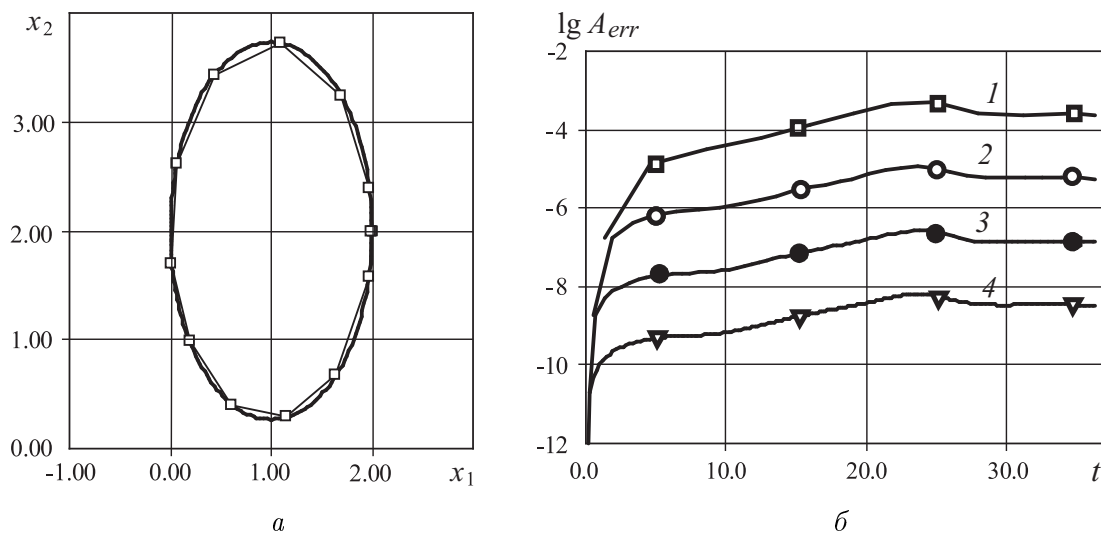


Рис. 2. Траектория решения в области A с указанием точек численного решения, найденных с точностью $Tol = 10^{-2}$ (a), и порядок абсолютной погрешности $\lg A_{err}$ численного решения на траектории одного цикла при разных заданных значениях точности: $Tol = 10^{-2}$ (1), 10^{-4} (2), 10^{-6} (3), 10^{-8} (4)

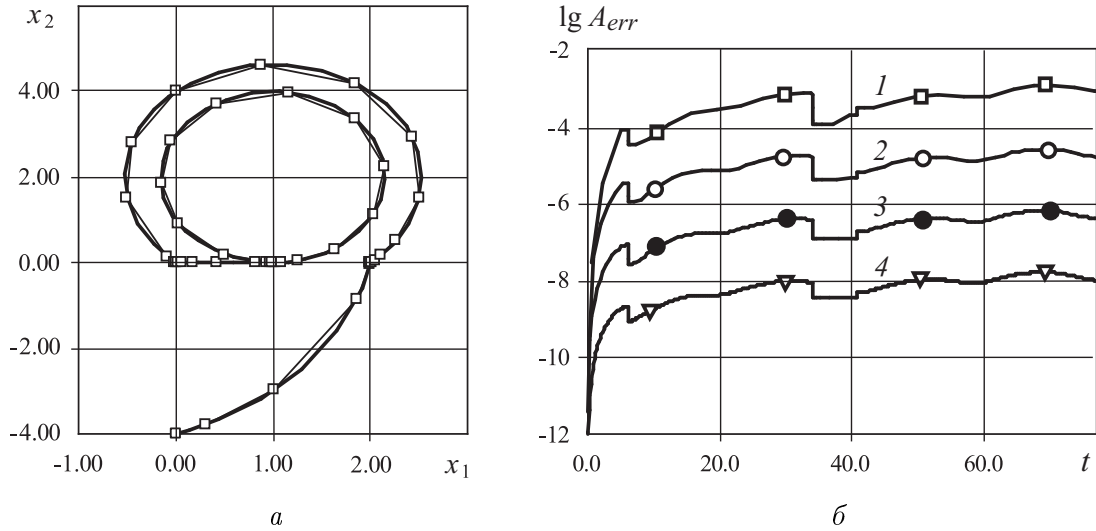


Рис. 3. Траектория решения в области B с указанием точек численного решения, найденных с точностью $Tol = 10^{-2}$ (а), и порядок абсолютной погрешности $\lg A_{err}$ численного решения на траектории одного цикла при разных заданных значениях точности: $Tol = 10^{-2}$ (1), 10^{-4} (2), 10^{-6} (3), 10^{-8} (4)

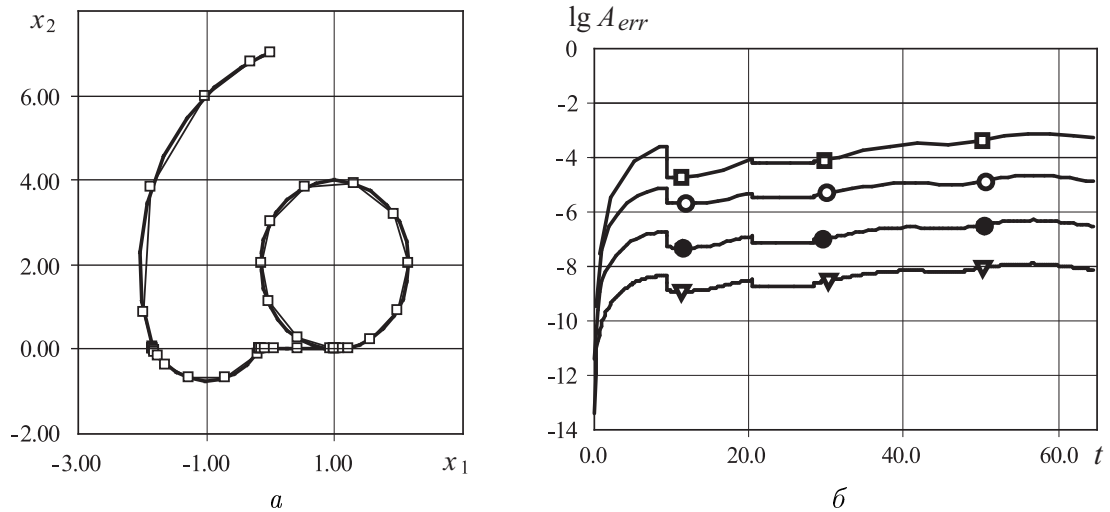


Рис. 4. Траектория решения в области C с указанием точек численного решения, найденных с точностью $Tol = 10^{-2}$ (а), и порядок абсолютной погрешности $\lg A_{err}$ численного решения на траектории одного цикла при разных заданных значениях точности: $Tol = 10^{-2}$ (\square), 10^{-4} (\circ), 10^{-6} (\bullet), 10^{-8} (∇)

случае. Тем не менее мы сочли нужным включить эту траекторию в вычислительный эксперимент. Результаты (рис. ??, б) показали сходимость численного решения к точному при уменьшении величины заданной точности Tol .

Поведение алгоритма можно проследить на графиках изменения количества шагов интегрирования вдоль траектории в областях B и C (рис. ??). Заметим, что рост количества шагов происходит вблизи поверхности разрыва: в первый раз — при пересечении поверхности, во второй раз — при входе в скользящий режим и в третий раз — при выходе из него. Увеличение заданной точности требует больших затрат на вычисление, поэтому происходит рост количества шагов интегрирования.

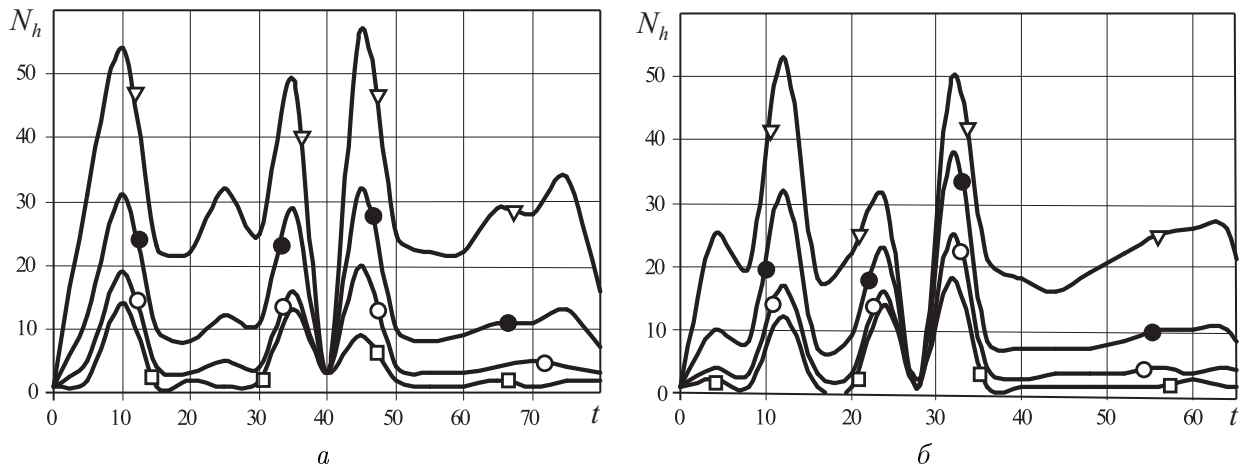


Рис. 5. Изменение количества шагов интегрирования N_h вдоль траектории в областях B (а) и C (б) при разных заданных значениях точности: $Tol = 10^{-2}$ (\square), 10^{-4} (\circ), 10^{-6} (\bullet), 10^{-8} (∇)

Вычислительные затраты алгоритма для нахождения численного решения адекватно оценить довольно сложно. Во-первых, вычислительные затраты численного метода решения ОДУ традиционно оценивают по количеству вычислений правых частей уравнения. Эта величина хорошо коррелирует с реальными вычислительными затратами для одно- и многошаговых методов, но в представленном алгоритме присутствуют итерационные процедуры, не использующие вычисления правых частей, но требующие временных затрат на сходимость. Во-вторых, вычислительные затраты, которые можно определить экспериментально, зависят от конкретной реализации и от компьютера, используемого для расчетов. Поэтому конкретные временные характеристики, выраженные в секундах работы алгоритма, сильно меняются в зависимости от применяемых модели компьютера и программной реализации.

Логично предположить, что удачной оценкой вычислительных затрат алгоритма будет количество шагов метода Эйлера, которое можно выполнить за время работы на данном компьютере в той же программной реализации. Именно такой критерий оценки производительности был выбран. Результаты эксперимента приведены в таблице, где представлены следующие параметры: Tol — заданная точность, N_h — количество принятых шагов метода для нахождения решения на заданном интервале, N_f — количество выполненных вычислений правых частей, T_c — время (в секундах), потраченное на вычисление в данной реализации, N_e — количество шагов метода Эйлера, которое можно выполнить за время T_c , V_h — коэффициент увеличения шагов интегрирования (по отношению к предыдущей строке в таблице), P — оценка порядка метода. Отметим, что для области A результаты эксперимента приведены для 10 циклов траектории, поскольку на одном цикле время точно измерить сложно — в этом случае требуемое время составляет меньше 0.1 с. Для областей B и C приведено значение времени вычисления одной траектории.

Проведенные вычисления показали, что среднее количество вычислений правых частей ОДУ на один принятый шаг интегрирования для области A равно 11.98, B — 13.88, C — 14.58. Коэффициент уменьшения шага интегрирования при увеличении точности в 10 раз не превышает значения 1.629, что для траектории A дает минимальный порядок метода 4.717. Для траектории B наибольший коэффициент уменьшения шага составил 1.524, а минимальный порядок — 5.461, для траектории C — соответственно 1.522

Оценка вычислительных затрат и порядка точности алгоритма

Tol	N_h	N_f	T_c	N_e	V_h	P
<i>Область А</i>						
10^{-2}	108	1287	0.261	1569	—	—
10^{-3}	174	2079	0.411	3777	1.615	4.801
10^{-4}	282	3375	0.480	4793	1.623	4.752
10^{-5}	459	5499	0.691	7899	1.629	4.717
10^{-6}	733	8787	1.202	15421	1.598	4.913
10^{-7}	1177	14127	1.703	22796	1.608	4.849
10^{-8}	1879	22551	2.593	35896	1.596	4.923
10^{-9}	2992	35895	4.096	58020	1.592	4.954
10^{-10}	4757	57075	6.489	93245	1.590	4.956
<i>Область В</i>						
10^{-2}	58	883	0.451	4366	—	—
10^{-3}	77	1169	0.461	4513	1.324	8.207
10^{-4}	105	1538	0.551	5838	1.316	8.393
10^{-5}	149	2105	0.641	7163	1.369	7.337
10^{-6}	217	3006	0.791	9371	1.428	6.463
10^{-7}	315	4230	0.902	11005	1.407	6.741
10^{-8}	472	6178	1.222	15715	1.461	6.079
10^{-9}	719	9199	1.572	20876	1.489	5.784
10^{-10}	1114	14023	2.103	28683	1.524	5.461
<i>Область С</i>						
10^{-2}	61	1039	0.410	3763	—	—
10^{-3}	77	1263	0.491	4955	1.216	11.794
10^{-4}	100	1589	0.610	6707	1.258	10.028
10^{-5}	138	2100	0.601	6574	1.322	8.258
10^{-6}	181	2545	0.621	6869	1.212	11.981
10^{-7}	265	3636	0.812	9680	1.429	6.454
10^{-8}	392	5207	1.061	13345	1.432	6.412
10^{-9}	593	7678	1.332	17334	1.457	5.929
10^{-10}	922	11689	1.833	24709	1.522	5.479

и 5.579. Этот результат показывает, что добавленные процедуры поиска точек пересечения траектории с границей и точек выхода из скользящего режима не понизили точность метода в целом и увеличили временные затраты не более чем на 24 %. Кроме того, условие оценки погрешности на границе областей выбрано слишком сильным. Как показывает эксперимент, порядок метода на низкой заданной точности слишком велик и опускается до 5.461 на высокой заданной точности. Это означает, что условие оценки погрешности требует корректировки в сторону ее увеличения, что также позволит уменьшить вычислительные затраты на низкой заданной точности.

Таким образом, в представленной работе описан алгоритм численного решения системы ОДУ с правой частью, которая терпит разрыв на гладкой поверхности. В ходе реализации алгоритма были решены задачи по нахождению точки входа траектории численного решения на поверхность разрыва, определению дальнейшего поведения решения, вычислению решения в скользящем режиме, нахождению точки выхода со скользящего режима. На примере численного решения одной системы показана сходимость предложенного численного метода.

Список литературы

- [1] Айзерман М.А., Пятницкий Е.С. Основы теории разрывных систем. I // Автоматика и телемеханика. 1974. № 7. С. 33–47.
- [2] Айзерман М.А., Пятницкий Е.С. Основы теории разрывных систем. II // Там же. 1974. № 8. С. 39–61.
- [3] Матросов И.В. О единственности справа решений невырожденных алгебро-дифференциальных уравнений с разрывами // Там же. 2007. № 1. С. 11–19.
- [4] Уткин В.И. Скользящие режимы в задачах оптимизации и управления. М.: Наука, 1981. 368 с.
- [5] Di Bernardo M., Johansson K.H., Vasca F. Self-oscillations and sliding in relay feedback systems: Symmetry and bifurcations // Intern. J. Bifurcation and Chaos. 2001. Vol. 11, No. 4. P. 1121–1140.
- [6] Di Bernardo M., Kowalczyk P., Nordmark A. Bifurcations of dynamical systems with sliding: Derivation of normal-form mappings // Phys. D. 2002. Vol. 170. P. 175–205.
- [7] Gear C.W., Osterby O. Solving ordinary differential equations with discontinuities // ACM Trans. Math. Software. 1984. Vol. 10. P. 23–44.
- [8] Gouze J.-L., Sari T. A class of piecewise linear differential equations arising in biological models // Dynamical Systems. 2002. Vol. 17. P. 299–319.
- [9] Johansson K.H., Barabanov A.E., Astrom K.J. Limit cycles with chattering in relay feedback systems // IEEE Transactions on Automatic Control. 2002. Vol. 47. P. 1414–1423.
- [10] Kowalczyk P., di Bernardo M., Champneys A.R. et al. Two-parameter nonsmooth bifurcations of limit cycles: Classification and open problems // Intern. J. Bifurcat. Chaos. 2006. Vol. 16. P. 601–629.
- [11] Kuznetsov Yu.A., Rinaldi S., Gagnani A. One-parameter bifurcations in planar Filippov systems // Ibid. 2003. Vol. 13. P. 2157–2188.
- [12] Leine R.I., van Campen D.H., van de Vrande B.L. Bifurcations in nonlinear discontinuous systems // Nonlinear Dynamics. 2000. Vol. 23. P. 105–164.
- [13] Malmborg J., Bernhardsson B. Control and simulation of hybrid systems // Commun. in Nonl. Sci. and Numerical Simulat. 1997. Vol. 30. P. 337–347.
- [14] Di Bernardo M., Budd C.J., Champneys A.R., Kowalczyk P. Piecewise-smooth Dynamical Systems. Theory and Applications. Applied Mathematical Sciences. Berlin: Springer-Verlag, 2008.
- [15] Zhusubaliyev Z., Mosekilde E. Bifurcations and Chaos in Piecewise-Smooth Dynamical Systems. Singapore: World Scientific, 2003.
- [16] Филиппов А.Ф. Дифференциальные уравнения с разрывной правой частью. М.: Наука, 1985. 224 с.
- [17] Новиков Е.А., Шорников Ю.В. Численное моделирование гибридных систем методом Рунге—Кутты второго порядка точности // Вычисл. технологии. 2008. Т. 13, № 2. С. 99–105.
- [18] Cash J.R., Karp A.H. A variable order Runge-Kutta method for initial value problems with rapidly varying right-hand sides // ACM Trans. Math. Software. 1990. Vol. 16, No. 3. P. 201–222.

- [19] Dieci L., Lopez L. Sliding Motion in Filippov Differential Systems: Theoretical Results and A Computational Approach. <http://www.math.gatech.edu/dieci/preps/DL-Fili.pdf> (01.03.2009).
- [20] Piironen P.T., Kuznetsov Yu.A. An event-driven method to simulate Filippov systems with accurate computing of sliding motions // ACM Trans. Math. Software. 2008. Vol. 34, No. 13. P. 1–24.
- [21] Shampine L.F., Thompson S. Event location for ordinary differential equations // Comput. Math. with Appl. 2000. Vol. 39. P. 43–54.
- [22] Guglielmi N., Hairer E. Computing breaking points in implicit delay differential equations // Advances in Comput. Math. 2008. Vol. 29, No. 3. P. 229–247.
- [23] Хайпер Э., Нёрсетт С., Ваннер Г. Решение обыкновенных дифференциальных уравнений. Нежесткие задачи. М.: Мир, 1990. 512 с.
- [24] Хайпер Э., Ваннер Г. Решение обыкновенных дифференциальных уравнений. Жесткие и дифференциально-алгебраические задачи. М.: Мир, 1999. 685 с.
- [25] Gupta G.K., Sacks-Davis R., Tescher P.E. A review of recent developments in solving ODEs // ACM Comput. Surveys. 1985. Vol. 17, No. 1. P. 5–47.
- [26] Park T., Barton P.I. State event location in differential-algebraic models // ACM Trans. on Modeling and Comput. Simulat. 1996. Vol. 6, No. 2. P. 137–165.
- [27] Berzins M., Brankin R.W., Gladwell I. The stiff integrators in the NAG library // ACM SIGNUM Newsletter. 1988. Vol. 23, No. 2. P. 16–23.
- [28] Enright W.H., Pryce J.D. Two FORTRAN packages for assessing initial value methods // ACM Trans. Math. Software. 1987. Vol. 13, No. 1. P. 1–27.
- [29] Kamel M.S., Enright W.H., Ma K.S. ODEXPERT: An expert system to select numerical solvers for initial value ODE systems // Ibid. 1993. Vol. 19, No. 1. P. 44–62.
- [30] Shampine L.F., Reichelt M.W. The MATLAB ODE suite // SIAM J. Sci. Comput. 1997. Vol. 18. P. 1–22.
- [31] Коробицын В.В., Фролова Ю.В. Алгоритм численного решения дифференциальных уравнений с разрывной правой частью // Математические структуры и моделирование. Омский гос. ун-т, 2005. Вып. 15. С. 46–54.
- [32] Коробицын В.В., Маренич В.Б., Фролова Ю.В. Исследование поведения явных методов Рунге—Кутты при решении систем обыкновенных дифференциальных уравнений с разрывной правой частью // Математические структуры и моделирование. Омский гос. ун-т, 2007. Вып. 17. С. 19–25.
- [33] Коробицын В.В., Фролова Ю.В., Маренич В.Б. Алгоритм численного решения кусочно-спшитых систем // Вычисл. технологии. 2008. Т. 13, № 2. С. 70–81.

Поступила в редакцию 1 апреля 2009 г.