УДК 519.622:004.8.023

Решение задачи Коши для обыкновенных дифференциальных уравнений методом генетического программирования

Сергей В. Бураков*
Институт математики,
Сибирский федеральный университет,
Свободный, 79, Красноярск, 660041,
Россия

Евгений С. Семенкин†
Сибирский государственный аэрокосмический университет,
пр. им. газ. "Красноярский рабочий" , 31, Красноярск, 660014,
Россия

Получена 31.05.2010, окончательный вариант 25.07.2010, принята к печати 10.10.2010

Для решения задачи Коши предлагается использовать метод генетического программирования, который позволяет находить точное аналитическое решение, если оно существует, и приближенное аналитическое выражение в противном случае. Рассматриваются особенности решения задачи Коши этим методом. Представлены данные численных экспериментов, проведенных с использованием предложенного метода.

Ключевые слова: обыкновенные дифференциальные уравнения, задача Коши, численные методы, алгоритм генетического программирования.

Введение

Задачи, связанные с решением обыкновенных дифференциальных уравнений (ОДУ), встречаются во многих областях науки и техники. Существуют два основных метода решения: традиционный [1] и численный [2]. Традиционный метод точен, теоретически обоснован, результат решения удобен для анализа и дальнейшего использования. Однако традиционным методом можно решить далеко не каждую задачу Коши для ОДУ, особенно велико количество таких задач на практике. В этом случае ОДУ решают численно: строят разностную схему, доказывают сходимость, вычисляют на ЭВМ, получают результат – числовую таблицу значений функции-решения. Такой результат ограничивает возможности анализа и его применения, да и просто неудобен, а при доказательстве сходимости, исследовании на устойчивость возможно появление дополнительных ограничений.

В силу того, что каждый из методов решения ОДУ имеет "слабые" стороны, представляется логичным шагом разработать и исследовать алгоритм решения ОДУ, который мог бы объединить преимущества обоих подходов, т.е. алгоритм решения задачи Коши (для как

*burakov_krasu@mail.ru
†eugenesemenkin@yandex.ru

можно большего количества ОДУ) в символьном виде. Алгоритмы генетического программирования позволяют решить эту задачу.

Поэтому целью статьи является разработка, исследование и настройка алгоритма генетического программирования, способного получать точное решение, если его можно представить элементарными функциями, или приближенное символьное решение в противном случае.

1. Постановка задачи и метод решения

Рассмотрим задачу решения ОДУ в общем виде. Пусть дано обыкновенное дифференциальное уравнение в виде

$$F\left(x, y, y', y'', \ldots, y^{(n)}\right) = 0 \tag{1}$$

и начальные условия

$$y(x_0) = Y_0, y'(x_0) = Y_0', \ldots, y^{(n-1)}(x_0) = Y_0^{(n-1)}, \tag{2}$$

где $y(x)$ — искомое решение, $y', y', \ldots, y^{(n)}$ — производные функции $y(x)$; Y_0, Y_0', Y_0'', …, $Y_0^{(n-1)}$ — заданные вещественные константы. Требуется найти функцию $y(x)$, удовлетворяющую уравнению (1) и начальным условиям (2). В теории обыкновенных дифференциальных уравнений доказана теорема существования и единственности решения для уравнения n-го порядка [1]. Далее, если не указано другого, будем считать, что требования теоремы (в общем случае непрерывная дифференцируемость и принадлежность точки области) удовлетворены, следовательно, решение задачи существует и единственно.

Процесс получения решения поставленной задачи в символьном виде можно рассматривать как сложную оптимизационную процедуру отыскания наименьшего значения функции ошибки на множестве символьных выражений, где глобальный оптимум равен нулю и достигается на точном символьном выражении, представляющем истинное решение задачи Коши. Для поиска символьного представления решения целесообразно проводить преобразования бинарных деревьев, представляющих собой математические функции. Поэтому для решения поставленной задачи удобно использовать алгоритм генетического программирования (ГП) [3], являющийся стохастической процедурой, имитирующей процессы эволюции. Эволюционные алгоритмы вообще и генетическое программирование в частности интуитивно понятны и легко программируемы, однако наряду с этим они служат мощным и гибким инструментом решения сложных задач моделирования и оптимизации, позволяющим осуществлять тонкую настройку параметров для поиска нужного решения, удовлетворяющего нескольким критериям.

2. Алгоритм генетического программирования

Для разработки алгоритма нужно подробнее определить общую концепцию работы алгоритма: данные, искомый результат, критерий остановки. Входные данные, как и в теоретической задаче, — это уравнение (символьное выражение) и набор начальных данных (вещественный вектор), в зависимости от порядка ОДУ. Искомым решением задачи будем считать символьную запись точного решения (с точностью до элементарных преобразований) или приближенное (с наибольшей возможной точностью) символьное выражение

в случае, когда решение нельзя представить элементарными функциями. Решение задачи должно удовлетворять уравнению и начальным условиям с малой погрешностью. Решение-кандидат называется индивидом, множество решений — популяцией, а операторы преобразования решений — это селекция, скрещивание, мутация. Способ вычисления погрешности не является принципиальным. Значение функции ошибки преобразуется в число, называемое пригодностью индивида (чем меньше ошибка, тем более пригоден индивид). В рамках данной задачи символьное представление решения задано бинарным деревом, состоящим из элементов функционального множества $(+, -, *, /, \sin, \cos, \exp, \log)$, и элементов терминального множества (термов x, y, а также вещественных коэффициентов). Очевидно, что бинарное дерево однозначно представляет символьную запись решения ОДУ, обратное, вообще говоря, неверно. Для решения общих задач терминальное и функциональное множества могут быть дополнены.

Упрощенно схему алгоритма можно представить [3] состоящей из следующих основных шагов, каждый из которых будет подробно описан ниже:

1. Инициализация начальной популяции (случайным образом).
2. Оценка пригодности каждого индивида.
3. Адаптация индивидов.
4. Применение генетических операторов к имеющимся индивидам для получения новой популяции.
5. Если критерий остановки не выполнен, то переход к шагу 2.

Здесь и далее бинарное дерево, представляющее решение ОДУ, будем называть «решение» в контексте теории ОДУ или «индивид» в контексте теории эволюции, так же как связки "погрешность" – "пригодность" (хотя эти величины взаимно обратные), "оптимизация" - "адаптация".

На первом этапе работы алгоритма строится популяция индивидов. Каждый индивид — бинарное дерево. До заданной пользователем глубины дерева решений случайным образом в каждый из узлов дерева записываются элементы функционального или терминального множеств. Вещественные коэффициенты (с одинаковой вероятностью относительно выбора термов) задаются случайно из определенного пользователем интервала.

На втором этапе каждый индивид популяции оценивается, т.е. вычисляется его пригодность. Пригодность индивида можно отождествлять с величиной, обратной погрешности (в соответствии с теорией эволюции предпочтительными должны быть более пригодные индивиды). Функция пригодности может вычисляться многими способами, хотя в ней так или иначе должна присутствовать погрешность относительно уравнения и начальных условий. Например, можно вычислять ее следующим образом:

$$Fit(P_k) = \frac{1}{1 + E(P_k) + K1 \cdot D(P_k)}; \tag{3}$$

$$E(P_k) = \frac{\sqrt{\sum_{i=1}^{N} |P_k(x_i)|}}{N} + K2 \cdot \sum_{j=0}^{n-1} \left| y^{(j)}(x_0) - Y_0^{(j)} \right|, \tag{4}$$

где $Fit(P_k)$ — значение функции пригодности k-го индивида, $E(P_k)$ — ошибка аппроксимации, вычисляемая по всем точкам выборки (N — объем выборки), K1 — коэффициент штрафа за сложность дерева, $D(P_k)$ — число вершин дерева P_k, K2 — коэффициент штрафа за начальные условия (n — количество начальных условий), $Y_0^{(j)}$ — заданные начальные

условия, $y^{(j)}(x_0)$ — производная j-го порядка в точке x_0 $(y^{(0)}(x_0) = y(x_0))$, $|P_k(x_i)|$ — отклонение (например, среднеквадратическое) функции F из выражения (1) от нуля, получаемое при подстановке решения P_k в ОДУ в выбранных точках (x_i) — точки из интервала могут быть выбраны равномерно, случайно или каким-либо специальным образом. Здесь при вычислении ошибки соответствия решения ОДУ требуется вычисление производных в точках выборки. Возможны два способа вычисления: численный (с определенным порядком малости) и аналитический — построить дерево производной функции и вычислить значения этой функции в точках выборки. Так как дифференцирование (в отличие от интегрирования) проводится всегда по строгим правилам, аналитически вычислить производную не представляет особого труда. Нетрудно заметить, что получение численной оценки производной требует меньше машинной памяти и времени, однако при вычислении производных более высоких порядков (чем первый, второй) накапливается погрешность аппроксимации производных и вычислительная погрешность, что ведет к искажению информации, поэтому метод вычисления производных должен выбираться с учетом решаемой задачи.

На третьем этапе производится адаптация каждого индивида из популяции. Индивид, лучше адаптирующийся в условиях поставленной задачи, имеет более высокую пригодность, а следовательно, более высокую вероятность быть отобранным для порождения потомков. С точки зрения теории ОДУ адаптация индивида означает изменение его частей так, чтобы решение имело меньшую погрешность, то есть означает оптимизацию решения — изменение вещественных коэффициентов и/или функционального набора бинарного дерева. Если зафиксировать все вершины дерева, кроме какого-то одного узла, то его (дерево) можно рассматривать как функцию одной переменной. Теория оптимизации содержит много методов поиска экстремумов. Предлагается воспользоваться методом Хука-Дживса, отличающимся простотой и эффективностью [4]. Бинарное дерево (решение ОДУ) может в общем случае содержать много вещественных коэффициентов и узлов-операций, это делает целесообразным при оптимизации проведение лишь небольшого числа итераций. Для ускорения поиска решения можно варьировать и функциональный набор, содержащийся в дереве.

На четвертом этапе применяются генетические операторы: селекция, рекомбинация (скрещивание), мутация. Селекция — оператор отбора индивидов с наибольшей пригодностью. Основная идея состоит в том, что более пригодные индивиды имеют больше шансов стать родителями нового решения или быть отобранными в новую популяцию. Существуют разные виды селекции: пропорциональная, турнирная, ранговая и др. Скрещивание или клонирование (вероятности скрещивания и клонирования задаются) следуют за селекцией: два отобранных индивида либо скрещиваются, либо один из них клонируется в новую популяцию. При скрещивании два дерева обмениваются поддеревьями, отсеченными в случайно выбранных точках. Велика вероятность того, что из деревьев с высокой пригодностью при скрещивании получатся деревья, превосходящие по пригодности родительские. Оператор мутации состоит в случайном изменении одного или нескольких узлов в дереве и рассматривается как средство восстановления генетического разнообразия — случайным образом разбрасывает решения по поисковому пространству.

Далее проверяется условие остановки алгоритма: выполнено заданное количество вычислений (прошло заданное количество времени) или достигнута заданная точность аппроксимации. Если условие выполняется, задача считается решенной, иначе происходит переход ко второму этапу.

Отметим, что параметры, касающиеся алгоритма генетического программирования: ко-

эффициенты скрещивания, мутации, максимальная глубина дерева, размер популяции и др., можно не изменять от задачи к задаче — эти параметры настраиваются исходя из общей концепции (высокая надежность, быстрая сходимость и т.д.). Оставшиеся параметры: границы интервала, объем выборки, коэффициент штрафа за несоответствие начальным условиям, порядок аппроксимации производных и др., должны задаваться исходя из конкретной задачи.

3. Комбинация алгоритма генетического программирования и численного метода

Задачу отыскания решения задачи Коши в символьном виде можно решать, используя комбинацию численного метода и алгоритма генетического программирования. Для этого предварительно ОДУ решается численно, например, методом Рунге-Кутта с четвертым порядком точности [2]. Решение численным методом предполагает предварительное видоизменение уравнения в пригодную для решения форму, вследствие чего некоторые частные решения должны быть рассмотрены отдельно. Если уравнение (1) является уравнением первого порядка, преобразуем его, если это возможно, к виду, пригодному для итерационной процедуры численного метода: $\frac{dy}{dx} = f(x, y), \quad y(x_0) = Y_0, \quad x > x_0$. В случае наличия производных более высокого порядка такое уравнение сводится к системе аналогичных дифференциальных уравнений 1-го порядка. Если в общем виде приведение уравнения к приемлемому для решения виду невозможно, применяют частные варианты решения таких уравнений.

Результатом численного метода является числовая таблица, представляющая собой значения искомой функции в заданных точках. Эта таблица рассматривается как входные данные для алгоритма ГП, который в этом случае решает задачу символьной регрессии в обычном смысле, подбирая символьное выражение, наиболее точно описывающее числовые данные.

Такой подход позволяет ускорить процесс работы алгоритма ГП — нет необходимости вычислять производные в точках выборки. При этом теоретическая обоснованность метода (слабое место всех стохастических процедур) повышается — доказаны существование и единственность решения для результата, полученного методом Рунге-Кутта (а алгоритм ГП представляет собой процедуру поиска интерполирующей функции). Однако возникают недостатки, свойственные численным методам: не все уравнения можно решить таким образом, в процессе приведения часть решений теряется — их приходится рассматривать отдельно, в некоторых задачах не выполняется условие непрерывной зависимости от входных данных. Кроме того, возникает недостаток, связанный с комбинацией численного метода с алгоритмом ГП, — наложение вычислительных погрешностей (особенно для степенных и экспоненциальных функций).

4. Результаты численных экспериментов

В программной реализации алгоритма ГП для решения задачи Коши предусмотрены два набора настроек. Во-первых, это основные настройки, определяющие задачу: само уравнение (в символьном виде), границы интервала, выборка точек в нем, тестовая функция-ответ

(если есть), коэффициенты штрафа за сложность дерева (для получения наименьшей записи решения), за несоответствие начальным условиям, способ вычисления производных (и порядок аппроксимации, если выбрана численная оценка производной) — эти настройки зависят от конкретной поставленной задачи. Во-вторых, настройки алгоритма ГП: размер популяции, начальная и максимально возможная глубина дерева, вероятности скрещивания, клонирования, мутации — эти настройки определяются из потребности решить задачу с наименьшим временем решения или с наибольшей точностью и надежностью.

Тестирование программы проводилось следующим образом: запускались одновременно четыре программы-приложения с одинаковой задачей, задача считалась решенной, когда хотя бы одно из приложений выдавало искомый результат.

Рассмотрим тестовую задачу: $y'' = -2 \cdot \sin(x), \quad y(0) = 0.0, \quad y'(0) = 2.0$. В качестве функционального множества будем использовать только элементарные арифметические операции (+,-,*/), что существенно усложнит алгоритму ГП поиск решения. Результатом решения такой задачи служит выражение

$$y(x) = 1,9996118 \cdot x - 0,3330926297 \cdot x^3 + 0,01662624753 \cdot x^5 - 0,00038863441 \cdot x^7 + 0,4403 \cdot 10^{-5} \cdot x^9.$$

Легко видеть, что последнее выражение является представлением точного решения (y=2$\cdot \sin(x)$) с помощью формулы Тейлора с остаточным членом достаточно высокого порядка малости.

В ходе тестирования подхода были решены задачи Коши, взятые из сборника задач по дифференциальным уравнениям [5]. В большинстве случаев были автоматически получены точные символьные решения, в некоторых случаях для этого потребовалось приведение подобных и вычисление несложных символьных выражений. В табл. 1 представлены некоторые из решенных задач и полученные результаты.

Таблица 1. Примеры решенных задач

№	Уравнение F()=0	Начальные условия	Точное решение	Полученные алгоритмом решения
1	$x \cdot y + (x+1) \cdot y' = 0$	$1,000[0;3]$	$(x+1) \cdot e^{-x}$	$(x+1) \cdot \exp(-1 \cdot x)$
2	$x^3 \cdot (y' - x) - y^2 = 0$	$0,000078285$ $[0,01;0,9]$	$x^2 - x^2/(\log(x))$	$(x - x/(\log(x))) \cdot x$
3	$x \cdot y' - 2 \cdot y - 2 \cdot x^4 = 0$	$2,0000[-1;1]$	$x^2 + x^4$	$((x \cdot x) \cdot ((x \cdot x) + 1))$
4	$y' + y \cdot \operatorname{tg}(x) - \sec(x) = 0$	$1,0000[0;6]$	$\sin(x) + \cos(x)$	$(\sin(x) + \cos(x))$
5	$2 \cdot x \cdot (x^2 + y) - y' = 0$	$4,139327065$ $[-1,4;1,4]$	$e^{x^2} - x^2 - 1$	$\exp(x \cdot x) - (x \cdot x + 1)$
6	$x \cdot y' + (x+1) \cdot y$ $-3 \cdot x^2 \cdot e^{-x} = 0$	$2,718282[-1;5]$ $4,126403[-1;5]$	$x^2/(e^{-x})$ $xy = (x^3+1)e^{-x}$	$(((x)/(\exp(x))) \cdot (x))$ $(1/x + x^2)/\exp(x)$
7	$(y - 1/x + y'/y) = 0$	$5,454545[1,2;10]$	$2 \cdot x/(x^2 - 1)$	$x \cdot 2/(x^2 - \sin(64,4))$
8	$(x^2 + 3 \cdot \log(y)) \cdot y$ $-x \cdot y' = 0$	$0,003606563$ $[-1,5;1,5]$	$e^{x^3 - x^2}$	$\exp((x \cdot x) \cdot$ $(\cos(-34,6) + x))$
9	$y'^2 + x \cdot y - y^2 - x \cdot y' = 0$	$0,135335[-2;2]$ $4,389056[-2;2]$	e^x $x - 1 + e^{-x}$	$\exp(x)$ $x - 1 + \exp(-x)$
10	$y'^2 - 2 \cdot x \cdot y' - 8 \cdot x \cdot x = 0$	$8,0[-2;2]$	$2 \cdot x^2$	$2 \cdot x^2 + \sin(\exp(-29))$
11	$y''^2 + y' - x \cdot y'' = 0$	$6,0 - 6,0[-1;5]$ $-4,33\ 4,0[-4;4]$	$x^2 - 4 \cdot x + 1$ $x^3/12 + 1$	$(1 + (x \cdot (x-4)))$ $1,00 + 0,083579 \cdot x^3$

В некоторых случаях решение задачи Коши не является единственным — например, для уравнения № 9 в табл. 1 при одинаковых начальных условиях существуют два решения $\exp(x)$ и $\exp(-x) + x - 5,25372)$, т.к. не выполнены требования теоремы единственности решения. Такие задачи в численных методах считаются некорректными и не могут быть решены в обычном режиме. Однако такие случаи не критические для предложенного алгоритма ГП. Алгоритм будет находить одно из решений (как правило, то, которое имеет более простую структуру — $\exp(x)$). Для того чтобы найти более сложное по структуре решение $(\exp(-x) + x - 5,25372)$, достаточно добавить в знаменатель функции вычисления пригодности (3) еще одно слагаемое, отвечающее за штраф при захвате известного решения, например, такое:

$$Pen(P_k) = K3 \cdot \frac{1}{\sum\limits_{j=1}^{M} \frac{\sqrt{\sum\limits_{i=1}^{N} (P_k(x_i) - F_j(x_i))^2}}{N}},$$

где M — количество частных решений (F_j), которые известны (или найдены на предыдущем этапе), и все они не являются искомым решением, K3 — коэффициент штрафа за приближение к известному решению.

Для сравнения работы прямого алгоритма ГП и алгоритма, комбинированного с численным методом, 10 задач были решены обоими методами. Задачи были приведены к виду, удобному для численных итераций, решены методом Рунге-Кутта 4-го порядка, а результат был записан в файл и на следующем этапе стал входом для алгоритма ГП. При этом каждая задача была решена по 20 раз каждым из методов, а результаты усреднены. В табл. 2 приведены получаемые решения, средний номер поколения ГП, на котором было впервые найдено это решение, средняя погрешность получаемых решений (знаменатель функции пригодности (3)), среднее время работы алгоритма.

Таблица 2. Результат решения задачи с 20-кратным прогоном

Полученное решение	№ поколения ГП	Погрешность	Время (мин.)
(((x)-(-1,000))/(exp(x)))	68	0,006	110
((((x)-(1,000))-(-2,000))/(exp(x)))	10	0,008	4
(((x)+(sin(-4,700)))/(exp(x)))	37	0,00722407	49
(((cos(0,000))+(x))/(exp(x)))	83	0,007	90
(((1,000)+(x))/(exp(x)))	10	0,006	4
(((((40,4)((1,1)+(x)))+(-4.1)))/(exp((3,7)+(x))))	265	0,0166029	748
(((x)-((-54,300)-(-53,300)))/(exp(x)))	180	0,008	311
(((x)+(cos(-37,700)))/(exp(x)))	19	0,00700115	13
(((1,000)+(x))/(exp(x)))	120	0,006	268
(((x)((sin(-29,800))+(x)))/((x)(exp(x))))	34	0,0139729	45
Среднее	82,6	0,008580102	164,2

Полученные решения можно разделить на группы: 1) точные решения или решения, приводимые к точным элементарными преобразованиями без округления (например, $(((x)-(-1,000))/(\exp(x)))$, $(((\cos(0,000)) + (x))/(\exp(x))))$; 2) условно-точные решения, приводимые к точным элементарными преобразованиями с использованием округления (например, $(((x) + (\sin(-4,700)))/(\exp(x))))$; 3) приближенные решения, т.е. решения, требующие более

сложных преобразований и/или имеющие сложную громоздкую, не приводимую к точному решению структуру дерева (например, (((((40, 4)·((1, 1)+(x)))+(−4, 1)))/(exp((3, 7)+(x))))). По результатам тестирования были вычислены средние показатели по всем задачам для каждой из групп при решении прямым и комбинированным алгоритмами ГП. Усредненные показатели представлены в табл. 3.

Таблица 3. Результаты решения 10 задач с 20-ти кратным прогоном каждой из них

	Прямой алгоритм	Комбинированный алгоритм
Средний процент точных решений	64	39
Средний процент условно-точных решений	22	7
Средний процент приближенных решений	14	54
Среднее количество поколений ГП	117	250
Среднее время работы (мин)	147	26
Количество точных решений для наиболее сложной задачи	2	0
Количество приближенных решений для наиболее сложной задачи	4	10

По результатам таблицы можно судить о том, что прямой алгоритм ГП более надежен и точен на всех задачах. Меньшую точность и надежность комбинированного алгоритма можно объяснить накапливаемой ошибкой: сначала ошибка накапливается при решении численным методом, а затем на нее накладывается ошибка при решении алгоритмом ГП. Большее среднее количество поколений ГП при решении комбинированным алгоритмом также можно считать следствием накопленной погрешности, так как она препятствует быстрому определению точного решения. Однако на некоторых задачах комбинированный алгоритм работает с сопоставимой точностью и надежностью и при этом тратит значительно меньше времени. Поэтому на настоящий момент однозначно сказать, что прямой алгоритм ГП является более предпочтительным, не представляется возможным.

Дополнительная возможность, которую предоставляет предложенный подход, состоит в возможности решать в символьном виде задачи Коши с ОДУ, где для получения ответа требуется взять "неберущиеся" интегралы. Например, решение прямым алгоритмом ГП задачи $y' - \frac{1}{\sqrt{2\pi}} \cdot e^{-x^2/2} = 0$, $x \in [0, 3]$, $y(0) = 0$, после элементарных преобразований имеет вид

$$\sin\left(-0,427 \cdot \left(\sin\left(\frac{0,125}{e^4} \cdot \sin(0,8 \cdot x) + 0,0107 \cdot x\right) - x\right)\right)$$

с погрешностью $6,65 \cdot 10^{-5}$.

Заключение

Предложен подход, основанный на алгоритме генетического программирования, позволяющий решать задачу Коши для обыкновенных дифференциальных уравнений в символьном виде. Разработанная программа, реализующая этот подход, дает возможность задавать ОДУ (в символьном виде) и начальные условия и решать задачу, получая точную формулу, если она существует, и приближенное символьное выражение в случае, когда решение нельзя выразить элементарными функциями, а также получать разложение искомой функции

в ряд Тейлора. Возможно вычисление производных точно, путем построения производной функции, или приближенно с помощью разностных схем. Процесс решения может реализовываться двумя способами. Получение решения прямым алгоритмом ГП требует больше времени, но точность результата и надежность получения искомого решения выше на большинстве задач. При решении комбинированным методом время работы значительно сокращается, однако на многих задачах одновременно с этим падает точность и надежность получения искомого решения. Поэтому предпочтение тому или иному методу решения должно отдаваться в зависимости от условий решения поставленной задачи.

Областью применения данного алгоритма может выступать любая отрасль науки и техники, так или иначе связанная с решением ОДУ. Однако такой подход нужно рассматривать как дополнительный инструмент в наборе уже имеющихся традиционных подходов к решению ОДУ. Нужно также отметить, что традиционные методы ориентируются на решение определенных известных типов уравнений, поэтому нельзя сравнивать их и алгоритм генетического программирования по точности или времени работы — очевидно, что традиционные методы будут точнее и быстрее, если решаемая задача удовлетворяет теоретическим положениям. Тем не менее, предложенный в данной работе подход заслуживает внимания в тех случаях, когда стандартные методы не могут дать желаемого результата.

Работа выполнена при финансовой поддержке АВЦП "Развитие научного потенциала высшей школы"(НИР 2.1.1/2710) и ФЦП "Научные и научно-педагогические кадры инновационной России" (НИР НК-136П/3).

Список литературы

[1] И.Г.Петровский, Лекции по теории обыкновенных дифференциальных уравнений, М., МГУ, 1984.

[2] А.А.Самарский, Теория разностных схем, М., Наука, 1989.

[3] J.R.Koza, Genetic Programming: On Programming Computer by Means of Natural Selection and Genetics, *Cambridge, MA, The MIT Press*, (1992), №1, 51-62.

[4] Б.Банди, Методы оптимизации (вводный курс), М., Радио и связь, 1988.

[5] А.Ф.Филиппов, Сборник задач по дифференциальным уравнениям, Москва–Ижевск, НИЦ "Регулярная и хаотическая динамика" , 2003.

Ordinary Differential Equations Cauchy Problem Solving with Genetic Programming Techniques

Sergey V. Burakov
Evgeny S. Semenkin

It is suggested to use genetic programming techniques for solving Cauchy problem that allows to get exact analytical solution if it does exist and approximate analytical expression otherwise. Features of solving process with this approach are considered. Results of numerical experiments are given.

Keywords: ordinary differential equations, Cauchy problem, numeric methods, genetic programming algorithm.