

УДК 519.6

О ВЫЧИСЛЕНИИ КОДИФФЕРЕНЦИАЛОВ

Т. А. Ангелов¹

В последние годы интенсивно развивается и применяется аппарат негладкого анализа, позволяющий исследовать задачи с негладкими функциями и находить решения задач недифференцируемой оптимизации. Для широкого класса недифференцируемых функций разработано квазидифференциальное и кодифференциальное исчисление. В статье описаны основные элементы математического и программного обеспечения для решения задач недифференцируемой оптимизации. Доказано несколько предложений, дополняющих существующий математический аппарат. Работа выполнена при финансовой поддержке РФФИ (код проекта 12–01–00752).

Ключевые слова: негладкий анализ, недифференцируемая оптимизация, кодифференциал.

Введение. Квазидифференцируемые и кодифференцируемые функции [1, 2] представляют собой достаточно широкий класс негладких функций. С помощью квазидифференциалов и кодифференциалов можно исследовать негладкую функцию в окрестности точки, получать локальные и глобальные условия оптимальности, а также строить эффективные алгоритмы решения оптимизационных задач. Существует исчисление негладкого анализа, позволяющее строить кодифференциалы суммы, произведения, частного и суперпозиции функций. Вычислив кодифференциал функции, можно эффективно решать многие прикладные задачи.

В настоящей статье приводится описание элементов программного обеспечения для решения задач оптимизации многомерных недифференцируемых функционалов.

1. Кодифференциал. Рассмотрим локально липшицевые функции, отображающие элементы пространства $X \subset \mathbb{R}^n$ в пространство \mathbb{R} . Как известно, эти функции являются почти везде дифференцируемыми за возможным исключением некоторого множества точек меры нуль. Для решения задач оптимизации рассматриваемых функций обычные градиентные методы неприменимы.

Определение 1 [1]. Пусть $f(x)$ — локально липшицевая функция, определенная на открытом множестве $X \subset \mathbb{R}^n$. Функция $f : X \rightarrow \mathbb{R}$ называется кодифференцируемой в точке $x \in X$, если существуют такие выпуклые компакты $\underline{d}f(x) \subset \mathbb{R}^{n+1}$ и $\overline{d}f(x) \subset \mathbb{R}^{n+1}$, что $f(x + \Delta) = f(x) + \Phi_x(\Delta) + o_x(\Delta)$, где

$$\begin{aligned} \Phi_x(\Delta) &= \max_{[a,v] \in \underline{d}f(x)} [a + (v, \Delta)] + \min_{[b,w] \in \overline{d}f(x)} [b + (w, \Delta)], \\ \frac{o_x(\alpha\Delta)}{\alpha} &\xrightarrow{\alpha \downarrow 0} 0 \quad \forall \Delta \in \mathbb{R}^n. \end{aligned} \tag{1}$$

Здесь $a, b \in \mathbb{R}$ и $v, w \in \mathbb{R}^n$. Если (1) имеет место равномерно по $\Delta \in S = \{\Delta \in \mathbb{R}^n \mid \|\Delta\| = 1\}$, то будем говорить, что функция f кодифференцируема в точке x равномерно по направлениям.

Пара $Df(x) = [\underline{d}f(x), \overline{d}f(x)]$ называется кодифференциалом функции f в точке x , $\underline{d}f(x)$ — гиподифференциалом, а $\overline{d}f(x)$ — гипердифференциалом.

Класс кодифференцируемых функций образует линейное пространство, а аппроксимация функции $f(x)$ в виде $F_x(\Delta) = f(x) + \Phi_x(\Delta)$ непрерывна по Δ .

Определение 2 [1]. Функция f называется непрерывно кодифференцируемой в точке x , если она кодифференцируема в некоторой окрестности точки x и если существует непрерывное (по Хаусдорфу [3]) в этой точке кодифференциальное отображение Df .

Класс непрерывно кодифференцируемых функций включает в себя суперпозиции непрерывно дифференцируемых функций и функций взятия поточечных минимума и максимума. Для непрерывно кодифференцируемой функции аппроксимация $F_x(\Delta)$ является непрерывной функцией и по x , что, в некотором смысле, роднит непрерывно кодифференцируемые функции с непрерывно дифференцируемыми, позволяя использовать это свойство в численных методах.

¹ Санкт-Петербургский государственный университет, факультет прикладной математики — процессов управления, Университетский просп., 35, 198504, Петергоф, Санкт-Петербург; аспирант, e-mail: angelov.t@gmail.com

Подробное изложение основных формул кодифференциального исчисления можно найти в [1]. Следует обратить внимание на следующие примеры непрерывно кодифференцируемых функций.

Пример 1. Пусть функция $f(x)$ непрерывно дифференцируема в окрестности точки $x \in X \subset \mathbb{R}^n$. Тогда $f(x + \Delta) = f(x) + (\nabla f(x), \Delta) + o_x(\Delta)$, где $\frac{o_x(\Delta)}{\|\Delta\|} \xrightarrow{\|\Delta\| \rightarrow 0} 0$ и $\nabla f(x)$ — градиент функции f в точке x . Заметим, что функция f непрерывно кодифференцируема в окрестности точки x равномерно по направлениям и в качестве кодифференциала можно взять $Df(x) = [\underline{d}f(x), \bar{d}f(x)]$, где

$$\underline{d}f(x) = \{[0, \nabla f(x)]\} \subset \mathbb{R}^{n+1}, \quad \bar{d}f(x) = \{0_{n+1}\} \subset \mathbb{R}^{n+1}. \quad (2)$$

Отображение $Df(x)$ непрерывно.

Пример 2. Пусть $x_0, x \in X \subset \mathbb{R}^n$. Рассмотрим функцию $y(x)$, непрерывно кодифференцируемую в окрестности точки x_0 .

Пусть $f(y)$ — дифференцируемая функция в окрестности точки $y_0 = y(x_0)$. Положим $\varphi(x) = f(y(x))$ для всех $x \in X$ и выразим кодифференциал $D\varphi(x_0) = [\underline{d}\varphi(x_0), \bar{d}\varphi(x_0)]$ через кодифференциал $Dy(x_0) = [\underline{d}y(x_0), \bar{d}y(x_0)]$. Воспользуемся теоремой 4.1 [1] (о кодифференцируемости суперпозиции). Получим, что функция $\varphi(x)$ непрерывно кодифференцируема в окрестности точки x_0 и ее кодифференциал равен $D\varphi(x_0) = f'(y_0)Dy(x_0)$.

Приведем простой пример негладкой функции, иллюстрирующий непрерывность кодифференциального отображения по определению 2.

Пример 3. Рассмотрим функцию $f(x) = |x|$, $x \in \mathbb{R}$, которую представим в виде $f(x) = \max\{x, -x\}$. Функция $f(x)$ является суперпозицией непрерывно дифференцируемых функций и функции взятия поточечного максимума, при этом $f(x)$ недифференцируема в точке $x_0 = 0$.

Пусть $f(x) = \max\{\varphi_1(x), \varphi_2(x)\}$, где $\varphi_1(x) = x$ и $\varphi_2(x) = -x$. Из примера 1 следует, что

$$D\varphi_1(x) = \left\{ \{(0, 1)^T\}, \{(0, 0)^T\} \right\}, \quad D\varphi_2(x) = \left\{ \{(0, 0)^T\}, \{(0, -1)^T\} \right\}.$$

Формула исчисления для функции $f(x) = \max_{i \in I} \{\varphi_i(x)\}$, $I = 1, 2$, имеет вид $Df(x) = [\underline{d}f(x), \bar{d}f(x)]$ [1], где

$$\underline{d}f(x) = \text{co} \left\{ \underline{d}\varphi_k(x) - \sum_{i \in I \setminus \{k\}} \bar{d}\varphi_i(x) + \{[\varphi_k(x) - f(x), 0]\} \mid k \in I \right\}, \quad \bar{d}f(x) = \sum_{i \in I} \bar{d}\varphi_i(x).$$

Вычислим пару гипо- и гипердифференциал $f(x)$ в точке $x_0 = 0$:

$$\begin{aligned} \underline{d}f(x_0) &= \text{co} \left\{ \begin{pmatrix} 0 \\ 1 \end{pmatrix} - \begin{pmatrix} 0 \\ -1 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix} \right\} = \text{co} \left\{ \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \end{pmatrix} \right\}, \\ \bar{d}f(x_0) &= \left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ -1 \end{pmatrix} \right\} = \left\{ \begin{pmatrix} 0 \\ -1 \end{pmatrix} \right\}. \end{aligned}$$

Следовательно, кодифференциал имеет вид $Df(x_0) = \left[\text{co}\{(0, 2)^T, (0, 0)^T\}, \{(0, -1)^T\} \right]$.

Заметим, что кодифференциал определяется не единственным образом. Например, можно его задать и в таком виде: $Df(x_0) = \left[\text{co}\{(0, 1)^T, (0, -1)^T\}, \{(0, 0)^T\} \right]$. В этой связи вводится понятие эквивалентных пар [1].

Пусть $[A, B]$ и $[C, D]$ — две пары выпуклых компактов.

Определение 3 [1]. Упорядоченные пары выпуклых компактов называются эквивалентными (обозначение: $[A, B] \approx [C, D]$), если $A - D = C - B$.

Последнее свойство будем использовать для нахождения расстояния Хаусдорфа для двух пар выпуклых компактов: $\text{dist}([A, B], [C, D]) = \rho_H(A - D, C - B)$.

Напомним метрику Хаусдорфа: $\rho_H(U, V) = \max \left\{ \max_{x \in U} \min_{y \in V} \|x - y\|, \max_{y \in V} \min_{x \in U} \|x - y\| \right\}$.

Определение 4. Отображение ψ , определенное на множестве Ω и имеющее компактные образы, называется непрерывным в точке $x \in \Omega$, если для любого $\varepsilon > 0$ найдется такое $\delta > 0$, что для $x' \in \mathcal{B}_\delta(x)$ выполняется неравенство $\rho_H(\psi(x), \psi(x')) < \varepsilon$. Здесь, как обычно, $\mathcal{B}_\delta(x)$ — шар радиуса δ с центром в точке x .

Зафиксируем $\varepsilon = \varepsilon_0 > 0$. Пусть $x_1 = \Delta$, $\Delta \neq 0$. Тогда

$$\underline{d}f(x_1) = \text{co} \left\{ \begin{pmatrix} \Delta - |\Delta| \\ 2 \end{pmatrix}, \begin{pmatrix} -\Delta - |\Delta| \\ 0 \end{pmatrix} \right\}, \quad \bar{d}f(x_1) = \left\{ \begin{pmatrix} 0 \\ -1 \end{pmatrix} \right\}.$$

Следовательно, кодифференциал имеет вид $Df(x_1) = \left[\text{co} \left\{ (\Delta - |\Delta|, 2)^T, (-\Delta - |\Delta|, 0)^T \right\}, \{(0, -1)^T\} \right]$.

Обозначим $A = \underline{d}f(x_0)$, $B = \bar{d}f(x_0)$, $C = \underline{d}f(x_1)$ и $D = \bar{d}f(x_1)$. Получим

$$\begin{aligned} U &= A - D = \text{co} \{ (0, 3)^T, (0, 1)^T \}, \\ V &= C - B = \text{co} \{ (\Delta - |\Delta|, 3)^T, (-\Delta - |\Delta|, 1)^T \} \quad \forall \Delta, \\ V_+ &= C - B = \text{co} \{ (0, 3)^T, (-2\Delta, 1)^T \}, \quad \text{если } \Delta > 0, \\ V_- &= C - B = \text{co} \{ (2\Delta, 3)^T, (0, 1)^T \}, \quad \text{если } \Delta < 0. \end{aligned}$$

Рассмотрим графическую интерпретацию последних множеств. На рис. 1 показывается, как V_+ стремится к U при $\Delta \downarrow 0$ и V_- стремится к U при $\Delta \uparrow 0$. Вычислим расстояние Хаусдорфа для случая $\Delta > 0$:

$$\begin{aligned} \rho_H(U, V_+) &= \max \left\{ \max \left\{ \min_{y \in V_+} \left\| \begin{pmatrix} 0 \\ 3 \end{pmatrix} - y \right\|, \min_{y \in V_+} \left\| \begin{pmatrix} 0 \\ 1 \end{pmatrix} - y \right\| \right\}, \max \left\{ \min_{x \in U} \left\| x - \begin{pmatrix} 0 \\ 3 \end{pmatrix} \right\|, \min_{x \in U} \left\| x - \begin{pmatrix} -2\Delta \\ 1 \end{pmatrix} \right\| \right\} \right\} = \\ &= \max \left\{ \max \left\{ 0, \frac{2\Delta}{\sqrt{\Delta^2 + 1}} \right\}, \max \{ 0, 2\Delta \} \right\} = \\ &= \max \left\{ \frac{2\Delta}{\sqrt{\Delta^2 + 1}}, 2\Delta \right\} = 2\Delta. \end{aligned}$$

Аналогично вычислим $\rho_H(U, V_-) = -2\Delta$ для $\Delta < 0$. Тогда при $0 < |\Delta| < \varepsilon_0/2$ выполняется $\rho_H(U, V) < \varepsilon_0$. Непрерывность по Хаусдорфу доказана.

Функция $f(x)$ квазидифференцируема (даже субдифференцируема) на \mathbb{R}^1 , причем в качестве квазидифференциала можно взять $Df(x) = [\underline{d}f(x), \bar{d}f(x)]$,

$$\bar{d}f(x) = \{0\}, \quad \underline{d}f(x) = \begin{cases} 1, & \text{при } x > 0, \\ -1, & \text{при } x < 0, \\ [-1, 1], & \text{при } x = 0. \end{cases}$$

Рассмотрим последовательность точек x , стремящуюся к 0 справа, тогда $U = [-1, 1]$, $V_+ = 1$ и

$$\rho_H(U, V_+) = \max \left\{ \max \left\{ \min_{y \in V_+} \| -1 - y \|, \min_{y \in V_+} \| 1 - y \| \right\}, \max \left\{ \min_{z \in U} \| z - 1 \| \right\} \right\} = \max \{ \max \{ 2, 0 \}, 0 \} = 2,$$

что означает разрыв квазидифференциального отображения в точке 0.

2. Решение задач недифференцируемой оптимизации. Для простоты изложения здесь и далее будет идти речь о минимизации функции. Многие методы недифференцируемой оптимизации являются обобщением классических градиентных методов. При реализации этих алгоритмов надо уметь эффективно выполнять следующие действия: вычисление кодифференциала (субдифференциала, квазидифференциала и др.) функции в заданной точке, проверка точки на оптимальность и нахождение направлений и величины шагов спуска. Ниже будут рассмотрены необходимые элементы для решения этих задач и сам метод кодифференциального спуска [1] для случая безусловной оптимизации.

2.1. Точный метод вычисления кодифференциала. В работе [4] предложен точный метод кодифференцирования. Вначале производится анализ и разложение заданной функции в виде “дерева”, а затем к “вершинам дерева” применяются формулы кодифференциального исчисления. Первый шаг состоит в том, чтобы разложить выражение на простейшие элементы с использованием операций \max , \min , $+$, $-$, $*$, $/$, $(\cdot)^n$. Здесь под простейшими элементами подразумеваются непрерывно-дифференцируемые функции, а $(\cdot)^n$ — операция возведения в степень натурального числа. Второй шаг представляет собой движение

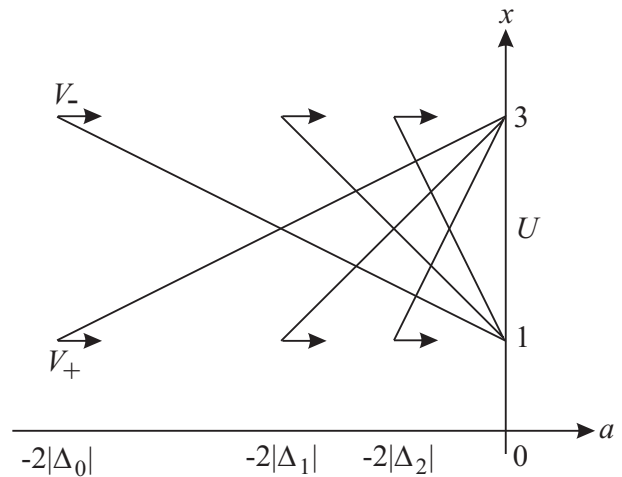


Рис. 1. Стремление V к U при $|\Delta_i| \rightarrow 0$

от вершин дерева с применением формул кодифференциального исчисления до получения кодифференциала исходной функции. Реализация, описанная в [4], имела некоторые слабые стороны: вычисление дифференциалов гладких функций вне алгоритма (встроенными библиотеками MatLab), неоптимальное использование памяти для сохранения ветвей дерева и значений кодифференциалов функций, невозможность вычисления некоторых суперпозиций функций (например, $f(x)^{g(x)}$ и $\min\left\{\cos\left(\max\{f(x), g(x)\}\right), 0\right\}$, где $f(x)$ и $g(x)$ — непрерывно-кодифференцируемые функции).

Дальнейшее усовершенствование метода точного вычисления кодифференциала предложено в настоящей статье. Его приложение позволяет устранить указанные трудности. Реализованный метод основывается на алгоритме стековой машины [5] для вычисления значения функции в точке. Ниже приведено краткое описание основных моментов модификации метода.

Программное обеспечение метода предлагает пользователю задать выражение исследуемой функции и точку, в которой требуется вычислить кодифференциал функции. Для задания выражения используется любая имеющая математический смысл композиция в инфиксной нотации всех алгебраических операций (+, −, *, /, ^), некоторых элементарных функций (тригонометрические, экспоненциальная, функции минимума и максимума), специальных символов, чисел десятичной системы и компонент вектора переменных.

Известно, что для гладких функций, таких как константа ($f_1(x) = \text{const}$) и переменная x^j ($f_2(x) = x^j$), $x = (x^1, \dots, x^n)^T \in X$, $j = 1, \dots, n$, можно аналитически определить кодифференциал с использованием примера 1. При этом обе функции $f_1(x)$ и $f_2(x)$ непрерывно кодифференцируемы в окрестности точки $x \in R^n$ равномерно по направлениям. Из (2) следует, что

$$Df_1(x) = \left[\{[0, 0_n]\}, \{0_{n+1}\} \right], \quad Df_2(x) = \left[\{[0, \nabla f_2(x)]\}, \{0_{n+1}\} \right], \quad (3)$$

где $\nabla f_2(x) = (0, 0, \dots, 0, \frac{1}{j}, 0, \dots, 0)^T$.

Для реализации первого этапа необходимо представить заданную функцию в виде композиции простейших функций, которыми являются константы и переменные. Это можно осуществить посредством перевода функции из инфиксной нотации в постфиксную, которая называется “обратной польской записью” [6]. На вершинах дерева разложения функции в ОПН (здесь и далее читать как “обратная польская нотация”) находятся только переменные и константы (рис. 2).

Рассмотрим разложение в ОПН на примере функции

$$f(x) = \min\{\max\{-1, x^3\}, 1\}, \quad x \in \mathbb{R}. \quad (4)$$

Перевод в постфиксную запись задается следующей строкой в ОПН: 1, −, x, 3, ^, max, 1, min.

Характерные особенности выражения в ОПН состоят в следовании символов операций за символами операндов и в отсутствии скобок. Вычисления в ОПН основаны на стеке; это означает, что операнды достаются из стека, а результат вычисления кладется обратно в стек. Напомним, что стек представляет собой способ хранения данных, основанный на принципе “последний пришел, первый вышел”. Обратная польская нотация обладает замечательным свойством — простотой анализа; поэтому ее реализации на компьютере требует мало ресурсов и выполняется быстро. Кроме того, ОПН явным образом задает порядок вычислений, избегая неоднозначностей, т.е. вычисление выражения может проводиться путем однократного просмотра.

На втором этапе обрабатываются только числовые значения функций и их кодифференциалов, которые хранятся в стеке. Алгоритм накопления кодифференциалов обходит строку ОПН и для каждого ее элемента производит соответствующие операции. Проведем анализ на следующем примере. Рассмотрим функцию (4) и найдем ее кодифференциал в точке $x_0 = 1$. Обозначим нижний ряд элементов дерева (рис. 2) через функции $\varphi_1(x) = 1$, $\varphi_2(x) = x$ и $\varphi_3(x) = 3$. При считывании второго элемента ОПН унарный минус, из стека выбрасывается последняя вложенная туда структура — значение функции $\varphi_1(x_0) = 1$ и ее кодифференциал $D\varphi_1(x_0) = \{[0_2], [0_2]\}$, которые умножаются на −1, а полученная структура возвращается на вершину стека. При просмотре пятого элемента ОПН (возведение в степень)

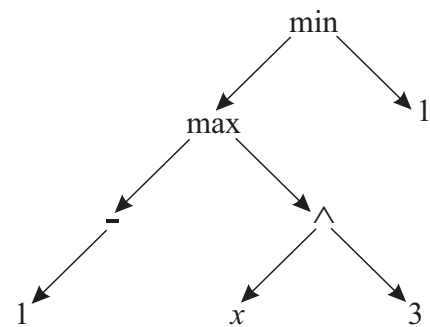


Рис. 2. Дерево функции $f(x) = \min\{\max\{-1, x^3\}, 1\}$

из стека выбрасываются последние две структуры — значение функции $\varphi_3(x_0) = 3$ и ее кодифференциал $D\varphi_3(x_0) = [\{0_2\}, \{0_2\}]$, $\varphi_2(x_0) = x_0$ и $D\varphi_2(x_0) = [\{(0, 1)^T\}, \{0_2\}]$. Исчисление производится так, как это указано в примере 2, и полученная структура возвращается на вершину стека. Единственные два случая, когда надо вводить в стек новые кодифференциалы (не производится предварительное выбрасывание элементов из стека), — это считывание констант или переменных, для которых выше были заданы кодифференциалы (3). Все остальные операции осуществляются с использованием формул кодифференциального исчисления для уже имеющихся кодифференциалов и значений функций.

По окончании рассмотрения строки ОПН в стеке остается только один элемент с кодифференциалом $Df(x_0) = \left[\text{co} \left\{ \begin{pmatrix} 0 \\ 3 \end{pmatrix}, \begin{pmatrix} -2 \\ 0 \end{pmatrix} \right\}, \text{co} \left\{ \begin{pmatrix} 0 \\ -3 \end{pmatrix}, \begin{pmatrix} 2 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \end{pmatrix} \right\} \right]$ и числовым значением $f(x_0) = 1$ в точке $x_0 = 1$.

2.2. Проверка условия минимума первого порядка. Пусть заданная функция f липшицева и непрерывно кодифференцируема на открытом множестве $X \subset \mathbb{R}^n$. Случай безусловной минимизации предполагает, что требуется найти минимум функции f во всем пространстве $X = \mathbb{R}^n$. Тогда

$$f(x + \Delta) = f(x) + \max_{[a,v] \in \underline{d}f(x)} [a + (v, \Delta)] + \min_{[b,w] \in \bar{d}f(x)} [b + (w, \Delta)] + o(\Delta),$$

где $\underline{d}f(x), \bar{d}f(x) \subset \mathbb{R}^{n+1}$ — выпуклые компакты, отображение $Df(x) = [\underline{d}f, \bar{d}f]$ непрерывно по Хаусдорфу и $\frac{o(\alpha\Delta)}{\alpha} \xrightarrow{\alpha \downarrow 0} 0$ для всех $\Delta \in \mathbb{R}^n$.

Необходимое условие минимума функции $f(x)$ в точке $x = x^*$ имеет вид

$$0_{n+1} \in \{ \underline{d}f(x^*) + [0, w] \} \quad \forall [0, w] \in \bar{d}f(x^*). \tag{5}$$

В контексте данной статьи выпуклые компакты $\underline{d}f(x)$ и $\bar{d}f(x)$ предполагаются многогранниками. При работе с этими множествами достаточно рассматривать их вершины. С целью оптимизации количества вычислений возникает задача нахождения вершин выпуклой оболочки конечного множества точек.

Алгоритмы нахождения выпуклых оболочек детально изложены в книге [7], в которой рассмотрен и метод Кейли “под, над” для многомерных пространств [8]. На основе последнего разработан алгоритм “QuickHull” [9], который используется в большинстве современных математических пакетов (MatLab, Maple). Для оптимизации вычислений необходимой информацией о выпуклой оболочке множества точек является множество вершин и все наборы симплицальных граней оболочки, которые получаются на выходе алгоритма “QuickHull”. Знание граней будет использовано ниже для решения задачи поиска расстояния от точки до выпуклого множества [10].

Обозначим $\bar{w} = [0, w] \in \bar{d}f(x^*)$ и $\{ \underline{d}f(x^*) + \bar{w} \} \equiv L_{\bar{w}}(x^*)$. Заметим, что необходимое условие минимума (5) эквивалентно соотношению

$$\min_{\bar{z} \in L_{\bar{w}}(x^*)} \|\bar{z}\| = 0 \quad \forall \bar{w} = [0, w] \in \bar{d}f(x^*). \tag{6}$$

Вычислим выпуклые оболочки $\underline{d}f(x^*)$ и $\bar{d}f(x^*)$. Выпуклое множество $L_{\bar{w}}(x^*)$ — это смещение $\underline{d}f(x^*)$ на вектор \bar{w} . Ясно, что каждая вершина множества $\underline{d}f(x^*)$ смещается на вектор \bar{w} и становится вершиной множества $L_{\bar{w}}(x^*)$, т.е. после вычисления выпуклой оболочки $\underline{d}f(x^*)$ нам становятся известны вершины и структура граней каждого множества $L_{\bar{w}}(x^*)$. Знание последних позволяет эффективно решать задачу нахождения расстояния от нуля до выпуклого множества $L_{\bar{w}}(x^*)$ (см. (6)), описанную в разделе 3.

Замечание 1. Не обязательно находить выпуклые оболочки компактов $\underline{d}f(x^*)$ и $\bar{d}f(x^*)$. Существуют алгоритмы [10–13], позволяющие решать задачу (6) и не требующие знания вершин множества $L_{\bar{w}}(x^*)$. Чтобы обосновать наш выбор, надо рассмотреть некоторые свойства объектов, с которыми проводятся вычисления. Одной из основных операций над парами выпуклых компактов является сложение (вычитание).

Пусть $[A, B]$ и $[C, D]$ — две пары выпуклых множеств. Тогда $[A, B] + [C, D] = [A + C, B + D]$, где оператор $+$ обозначает сложение по Минковскому, т.е. $A + B = \{x \mid x = a + b, a \in A, b \in B\}$. Каждая такая операция увеличивает количество векторов, описывающих выпуклые множества, со скоростью геометрической прогрессии. Чтобы как-то ограничить этот колоссальный рост данных, ставится задача “выброса” внутренних точек выпуклых множеств, так как вершины являются достаточными для задания выпуклой оболочки политопа. Уменьшение количества рассматриваемых точек сокращает последующие вычисления и приводит к уменьшению накапливания ошибок. В настоящей работе использован лучший

на данный момент реализованный алгоритм нахождения выпуклой оболочки в конечномерном пространстве “QuickHull”. На его применение уходит больше всего процессорного времени в сравнении с другими частями программы, а для размерностей $n > 9$ авторы [9] и вовсе не рекомендуют пользоваться этим алгоритмом.

2.3. Алгоритм кодифференциального спуска [1]. Зафиксируем любое $\mu > 0$. Выберем произвольное приближение $x_0 \in \mathbb{R}^n$. Пусть уже построено приближение $x_k \in \mathbb{R}^n$. Если в точке x_k выполнено условие (5), то x_k — точка минимума и процесс прекращается. В противном случае для каждого $\bar{w} \in \bar{d}_\mu f(x_k)$, где

$$\bar{d}_\mu f(x_k) = \left\{ \tilde{w} \in \bar{d}f(x_k) \mid \tilde{w} = (\omega, w), 0 \leq \omega \leq \mu \right\}, \tag{7}$$

находим

$$\min_{\bar{z} \in L_{\bar{w}}(x_k)} \|\bar{z}\| = \|\bar{z}_{k\bar{w}}\|; \tag{8}$$

здесь $\bar{z}_{k\bar{w}} = [\eta_{k\bar{w}}, z_{k\bar{w}}]$, $\eta_{k\bar{w}} \in \mathbb{R}$, $z_{k\bar{w}} \in \mathbb{R}^n$ и $L_{\bar{w}}(x_k) = \{\underline{d}f(x_k) + \bar{w}\}$. Подробное описание решения задачи (8) дается в разделе 3. Теперь для каждого $\bar{w} \in \bar{d}_\mu f(x_k)$ найдем

$$\min_{\alpha > 0} f(x_k - \alpha z_{k\bar{w}}) = f(x_k - \alpha_{k\bar{w}} z_{k\bar{w}}), \tag{9}$$

$$\min_{\bar{w} \in \bar{d}_\mu f(x_k)} f(x_k - \alpha_{k\bar{w}} z_{k\bar{w}}) = f(x_k - \alpha_{k\bar{w}_k} z_{k\bar{w}_k}). \tag{10}$$

Положим $x_{k+1} = x_k - \alpha_{k\bar{w}_k} z_{k\bar{w}_k}$. Далее продолжаем аналогично. В результате строим последовательность $\{x_k\}$, такую, что $f(x_{k+1}) < f(x_k)$.

Пример 3. Рассмотрим функцию (рис. 3)

$$f(x) = \max \left\{ \min \left\{ \max \left\{ -x + 8, \frac{3}{7}x - 2 \right\}, -\frac{2}{7}x + 8 \right\}, x - 28 \right\}, \tag{11}$$

где $x \in \mathbb{R}$. В точке $x_1 = 9$ условие (5) не выполняется и алгоритм находит два направления $-\frac{z_{k\bar{w}}}{\|z_{k\bar{w}}\|} = \pm 1$. Из рисунка видно, что функция в положительном направлении от точки x_1 сначала возрастает, а потом убывает, а в отрицательном убывает до достижения локального минимума.

Определение 5. Назовем $-z_{k\bar{w}} \in \mathbb{R}^n$ возможным направлением спуска, если в (8) $\|\bar{z}_{k\bar{w}}\| > 0$.

Замечание 2. Если в (7) задать $\mu < +\infty$, то некоторые возможные направления спуска не будут рассматриваться в задачах (8)–(10). Алгоритм можно назвать усеченным по гипердифференциалу. Такое усечение может привести к более быстрой остановке алгоритма. Заметим, что при $\mu = +\infty$ будут использованы все вершины гипердифференциала.

Замечание 3. Кодифференциал является инструментом “глобальной” оптимизации, а условие (5) — следствие условия локального минимума квазидифференцируемых функций [1, 2]. Если в точке x_k выполняется условие (5), то следует проверить все возможные направления спуска, т.е. решить задачи (8)–(10) при $\mu = +\infty$. Если $f(x_{k+1}) \geq f(x_k)$, то алгоритм прекращает свою работу.

Чтобы пояснить замечание 3, вернемся к рассмотрению функции (11). Пусть $\mu = +\infty$. В точке $x_2 = 7$ кодифференциал функции $f(x)$ имеет вид $Df(x_2) = [\underline{d}f(x_2), \bar{d}f(x_2)]$, где

$$\underline{d}f(x_2) = \text{co} \left\{ \begin{pmatrix} -27 \\ 2.714 \end{pmatrix}, \begin{pmatrix} -27 \\ 1.286 \end{pmatrix}, \begin{pmatrix} 0 \\ 1.428 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \end{pmatrix} \right\}, \quad \bar{d}f(x_2) = \text{co} \left\{ \begin{pmatrix} 5 \\ -1.714 \end{pmatrix}, \begin{pmatrix} 5 \\ -0.286 \end{pmatrix}, \begin{pmatrix} 0 \\ -1 \end{pmatrix} \right\}.$$

Существует единственная вершина из гипердифференциала с первым нулевым элементом $\bar{w} = [0, w]$, $\bar{w} \in \bar{d}f(x_2)$, $\bar{w} = (0, -1)^T$. Условие минимума (5) выполняется, так как

$$0_2 \in \left\{ \text{co} \left\{ (0, 0)^T, (0, 1.428)^T \right\} + (0, -1)^T \right\} = \left\{ \text{co} \left\{ (0, -1)^T, (0, 0.428)^T \right\} \right\}.$$

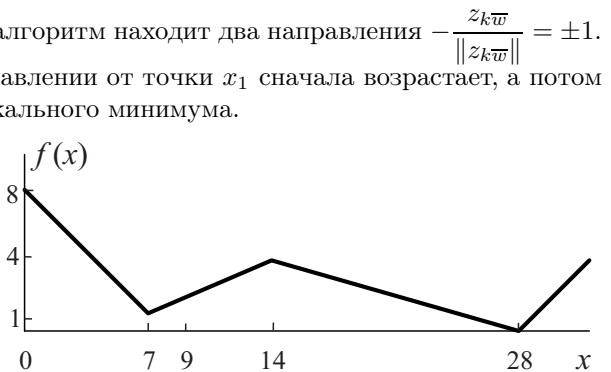


Рис. 3. График функции (11)

Решив задачу (8), находим возможное направление спуска $-\frac{z_{k\bar{w}}}{\|z_{k\bar{w}}\|} = 1$. Далее, решаем задачи (9), (10) и находим точку $x_3 = 28$. В точке x_3 условие минимума (5) выполняется и не существуют возможные направления спуска. Алгоритм останавливается.

При выборе метода одномерной минимизации для решения задачи (9) приходится учитывать тот факт, что после каждого вычисления кодифференциала функции в точке x_k находятся выпуклые оболочки компактов $\underline{d}f(x_k), \bar{d}f(x_k)$, что занимает очень много машинного времени. Поэтому количество шагов следует свести к минимуму. В программной реализации используется следующий алгоритм.

1. Интервал $[0, \alpha_{\max}]$ делится на $K \in \mathbb{N}$ равных частей длиной $h \in \mathbb{R}$, где $\alpha_{\max} \in \mathbb{R}$ — верхнее ограничение для α . Длина шага разбиения зависит от специфики функции. Для каждой точки $\alpha_i = ih, i = 0, 1, \dots, K$, вычисляется значение $f(\alpha_i)$ и берется α^* , для которой целевая функция достигла минимального значения. Можно подстраховаться и взять несколько наименьших значений $f(\alpha_i)$, при этом проследить, чтобы α_i не были соседними.

2. Рассматривается интервал $[\alpha^* - h, \alpha^* + h]$. Если $\alpha^* = \alpha_0$, то рассматривается интервал $[\alpha^*, \alpha^* + h]$. Если существуют такие $i, j = 0, \dots, K$, что $f(\alpha_i) \neq f(\alpha_j)$, то $\alpha^* - h, \alpha^*, \alpha^* + h$ называют “удачной тройкой чисел” [14], тем более что при достаточно малом h функция $f(\alpha_i)$ унимодальна на интервале $[\alpha^* - h, \alpha^* + h]$.

3. Используется метод золотого сечения для интервала $[\alpha^* - h, \alpha^* + h]$ до достижения нужной точности вычисления. В результате вычисляется минимальное значение функции $f\left(x_k - \alpha_{k\bar{w}} \frac{z_{k\bar{w}}}{\|z_{k\bar{w}}\|}\right)$ по рассматриваемому направлению и шаг спуска $\alpha_{k\bar{w}}$.

Задача (10) является простым сравнением уже найденных числовых значений функции в (9) по вершинам гипердифференциала \bar{w} , от которых получились возможные направления спуска (замечание 2).

Для создания эффективного программного обеспечения необходимо задать точность вычисления $c > 0$. Если $|\beta| < c, \beta \in \mathbb{R}$, то будем считать $\beta = 0$. При выборе точности вычислений нужно учитывать структуры мантисс использованных в программе переменных и значений функций.

3. Расстояние от начала координат до выпуклого многогранника. Пусть выпуклый многогранник $M \subset \mathbb{R}^{n+1}$ задается выпуклой оболочкой конечного набора точек M . Рассмотрим задачу поиска расстояния от начала координат до выпуклого многогранника M :

$$\min_{\psi \in \text{co } M} \|\psi\| = \|\bar{\psi}\|. \tag{12}$$

Пусть размерность многогранника M равна $n + 1, n \geq 1$. Тогда $\text{int co } M \neq \emptyset$.

Структура многогранника M определяется его границами. Поэтому все $\xi \in M$, для которых $\xi \in \text{int co } M$, не несут дополнительной информации и могут быть удалены из M . Такое удаление “лишних точек” облегчает работу с множеством M . После нахождения выпуклой оболочки M становится известной структура границы многогранника M , состоящая из множества граней $\{\hat{P}_l\} \subset \mathbb{R}^n, l \in J$. Пусть $\{P_k\} \subset \mathbb{R}^n, k \in I$ — множество симплицальных граней, т.е. каждая из них задана $n + 1$ вершинами. Тогда $\hat{P}_l = \bigcup_{k \in I_l} P_k, l \in J, I_l \subset I$. Если предположить, что $0_{n+1} \notin \text{co } M$, то задача (12) сводится к нахождению

$$\min_{k \in I} \min_{\psi \in P_k} \|\psi\| = \|\bar{\psi}\|. \tag{13}$$

Решение задачи (13) при $0_{n+1} \notin \text{co } M$ подробно описано в [15]. Этот метод находит точное расстояние от начала координат до границы многогранника M , если $0_{n+1} \in \text{int co } M$. Достоинством метода является возможность распараллеливания вычислений по граням P_k , а также простота реализации. Ниже будет показано, как использовать (13) для решения задачи (12).

Дадим краткое описание метода [15]. Для каждого $k \in I$ рассматривается $P_k = \text{co}\{X^1, X^2, \dots, X^{n+1}\}, \dim \text{aff } P_k = n, X^i \in \mathbb{R}^{n+1}, i = 1, \dots, n + 1$. Пусть векторы $\{X^i\}$ линейно независимы. Найдем решение системы линейных уравнений

$$\begin{pmatrix} (X^1, X^1) & \dots & (X^1, X^{n+1}) \\ (X^2, X^1) & \dots & (X^2, X^{n+1}) \\ \vdots & \ddots & \vdots \\ (X^{n+1}, X^1) & \dots & (X^{n+1}, X^{n+1}) \end{pmatrix} \begin{pmatrix} \bar{\lambda}_1 \\ \bar{\lambda}_2 \\ \vdots \\ \bar{\lambda}_{n+1} \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix},$$

где $\bar{\lambda}_1 + \bar{\lambda}_2 + \dots + \bar{\lambda}_{n+1} = \frac{1}{p}, \lambda_j = p\bar{\lambda}_j, j = 1, \dots, n + 1, p \in \mathbb{R}$. Если все $\lambda_1, \dots, \lambda_{n+1}$ неотрицательны, то

найденно расстояние $\|\bar{X}_k\| = \|\lambda_1 X^1 + \dots + \lambda_{n+1} X^{n+1}\|$ до данного симплекса P_k . Если существуют отрицательные $\lambda_{j_0} < 0$, $j_0 = 1, \dots, r$, $r \leq n$, то убираем соответствующие им X^{j_0} , решаем систему размерности $n + 1 - r$ и получаем ближайшую точку множества P_k к началу координат.

Теперь рассмотрим случай, когда векторы $\{X^i\}$ линейно зависимы. Тогда из $n + 1$ выбранных вершин будем рассматривать n вершин и начинать алгоритм заново. По окончании перебора всех $k \in I$ будет найдено минимальное расстояние до границы выпуклой оболочки $\min_{k \in I} \|\bar{X}_k\| = \|\bar{X}_{\min}\|$. Отделимость многогранника M от 0_{n+1} можно определить, если к описанному методу добавить несколько условий.

Замечание 4. Когда условие на линейную зависимость $\{X^i\} \subset P_k$ выполняется, то $0_{n+1} \in \text{aff } P_k$. Для проверки принадлежности 0_{n+1} к P_k решим систему

$$\alpha_2(X^2 - X^1) + \dots + \alpha_{n+1}(X^{n+1} - X^1) = -X^1, \quad \alpha_j \in \mathbb{R}, \quad j = 2, \dots, n+1. \quad (14)$$

Напомним, что $P_k = \text{co}\{X^1, \dots, X^{n+1}\}$ является симплексом и $\text{rank}(X^1, \dots, X^{n+1}) = n$. Тогда матрица $(X^2 - X^1, \dots, X^{n+1} - X^1)$ размерности $(n+1) \times n$ имеет ранг n . Заключаем, что решение переопределенной системы (14) существует и единственно. Если полученные α_j отвечают условиям $\alpha_j \geq 0$ и $\sum_{j=2}^{n+1} \alpha_j \leq 1$, то

$0_{n+1} \in P_k$, а тогда $0_{n+1} \in \text{co } M$ и $\bar{\psi} = 0_{n+1}$.

Обозначим через \tilde{J} множество индексов $\{l \in J \mid 0_{n+1} \in \text{aff } \hat{P}_l\}$.

Лемма 1. Если $0_{n+1} \notin \hat{P}_l$ для всех $l \in \tilde{J}$, то $0_{n+1} \notin \text{co } M$.

Доказательство. Рассмотрим грань \hat{P}_l , $l \in \tilde{J}$, выпуклого многогранника M , такую, что $0_{n+1} \notin \hat{P}_l$. Тогда существует соседняя грань \hat{P}_m , $m \in J \setminus \tilde{J}$, которая пересекает \hat{P}_l , а $\text{aff } \hat{P}_m$ делит $\text{aff } \hat{P}_l$ на две части: в одной находится \hat{P}_l , а в другой — 0_{n+1} . Тогда $\text{aff } \hat{P}_m$ отделяет многогранник M от нуля. Поэтому, если $0_{n+1} \notin \hat{P}_l$ для всех \hat{P}_l , $l \in \tilde{J}$, то $0_{n+1} \notin \text{co } M$. Что и требовалось доказать.

Обозначим через \tilde{I} множество индексов $\{k \in I \mid 0_{n+1} \in \text{aff } P_k\}$.

Следствие 1. Если $0_{n+1} \notin P_k$ для всех $k \in \tilde{I}$, то $0_{n+1} \notin \text{co } M$.

Пусть найдена единственная точка \bar{X}_{\min} на границе многогранника M . Вектор \bar{X}_{\min} принадлежит либо внутренности грани размерности n , либо ребру.

Теорема 1. Пусть вектор \bar{X}_{\min} принадлежит ребру выпуклого многогранника M и 0_{n+1} не принадлежит границе многогранника M . Тогда $0_{n+1} \notin \text{co } M$.

Доказательство. Пусть вектор \bar{X}_{\min} принадлежит ребру, которое является результатом пересечения двух граней \hat{P}_l и \hat{P}_m , $l, m \in J$. Пусть p_l и p_m — ортогональные проекции 0_{n+1} на $\text{aff } \hat{P}_l$ и $\text{aff } \hat{P}_m$ соответственно, $p_l \in \text{aff } \hat{P}_l$, $p_m \in \text{aff } \hat{P}_m$. Тогда $\|p_l\| \leq \|\bar{X}_{\min}\|$ и $\|p_m\| \leq \|\bar{X}_{\min}\|$. Однако вектор \bar{X}_{\min} не может быть одновременно ортогонален \hat{P}_l и \hat{P}_m . Тогда хотя бы один из векторов проекций меньше по норме, чем вектор к ближайшей точке многогранника. Пусть $\|p_m\| < \|\bar{X}_{\min}\|$, тогда $p_m \notin \hat{P}_m$. Из доказательства леммы 1 следует, что $p_m \notin \text{co } M$, где многогранник M есть компакт. Получена точка p_m , которая ближе к 0_{n+1} , чем \bar{X}_{\min} , и не принадлежит многограннику M . Следовательно, $0_{n+1} \notin \text{co } M$. Теорема 1 доказана.

Теорема 2. Пусть вектор $\bar{X}_{\min} \in \mathbb{R}^{n+1}$ принадлежит внутренности грани $\hat{P}_0 \subset \mathbb{R}^n$ выпуклого многогранника $M \subset \mathbb{R}^{n+1}$ и 0_{n+1} не принадлежит границе многогранника M . Если существует

$$a \in \left\{ y - \bar{X}_{\min} \mid y \in \text{co } M, y \notin \text{aff } \hat{P}_0 \right\},$$

для которого

$$(-\bar{X}_{\min}, a) < 0, \quad \text{то } 0_{n+1} \notin \text{co } M. \quad (15)$$

И наоборот, если

$$(-\bar{X}_{\min}, a) > 0, \quad \text{то } 0_{n+1} \in \text{co } M. \quad (16)$$

Доказательство. Покажем, что рассмотрение грани \hat{P}_0 является достаточным для определения положения точки 0_{n+1} . Пусть $\min_{l \in J \setminus \{0\}} \|\bar{Y}_l\| = b > \|\bar{X}_{\min}\|$, где $\bar{Y}_l \in \hat{P}_l$, $\bar{Y}_l = \min_{k \in I} \min_{\psi \in P_k} \|\psi\|$, J — множество всех граней $\text{co } M$. Из условия $\bar{X}_{\min} \in \text{int } \hat{P}_0$ следует, что существует $\varepsilon > 0$, такое, что $\varepsilon < b - \|\bar{X}_{\min}\|$. Тогда гипершар с центром 0_{n+1} и радиусом $\|\bar{X}_{\min}\| + \varepsilon$ пересекается только гранью \hat{P}_0 , которая разделяет этот

шар на две части, каждая из которых полностью лежит либо внутри многогранника M , либо снаружи, а принадлежность 0_{n+1} одной из частей определяется (15), (16). Теорема доказана.

Замечание 5. Пусть после перебора всех $P_k, k \in I$, найдены несколько разных векторов $\bar{X}_j, j = 1, \dots, r, r > 1$, для которых $\min_{k \in I} \|\bar{X}_k\| = \|\bar{X}_{\min}\| = \|\bar{X}_j\|, j = 1, \dots, r$. Тогда $0_{n+1} \in \text{co } M$. Этот факт — следствие существования единственной точки на выпуклом компакте, которая является ближайшей к точке вне компакта.

Данное условие отделимости нуля от выпуклого компакта неэффективно при реализации программного обеспечения из-за возможности проявления разных машинных ошибок для одного и того же вектора.

Замечание 6. Прежде чем использовать сложные алгоритмы решения задачи (12), проверяется случай, когда $0_{n+1} \in M$, т.е. $\bar{\psi} = 0_{n+1}$.

Замечание 7. Пусть размерность многогранника M равна $d_1 < n + 1$. Все вышесказанное остается справедливым, если перейти в пространство с размерностью, совпадающей с d_1 . Решение задачи (12) для $d_1 = 1$ и $d_1 = 0$ тривиально.

4. Пример [10]. Пусть $X = (x_1, x_2); f_1(X) = -5x_1 + x_2; f_2(X) = 4x_2 + x_1^2 + x_2^2; f_3(X) = 5x_1 + x_2; \varphi(X) = \max_{i \in \{1:3\}} f_i(X)$. Нужно найти $\varphi(X) \rightarrow \min_X$. Поскольку функции $f_i(X), i \in \{1, 2, 3\}$, выпуклы, то выпуклой будет и функция $\varphi(X)$.

$$\text{Введем обозначения: } \begin{cases} C_1 = \{X \mid f_1(X) = f_2(X)\} = \left\{ X \mid \left(x_1 + \frac{5}{2}\right)^2 + \left(x_2 + \frac{3}{2}\right)^2 = \frac{34}{4} \right\}, \\ C_2 = \{X \mid f_2(X) = f_3(X)\} = \left\{ X \mid \left(x_1 - \frac{5}{2}\right)^2 + \left(x_2 + \frac{3}{2}\right)^2 = \frac{34}{4} \right\}, \\ \bar{C} = \{X \mid f_1(X) = f_3(X)\} = \{X \mid x_1 = 0\}. \end{cases}$$

Нетрудно проверить, что (рис. 4)

$$\varphi(X) = \begin{cases} f_1(X), & \text{если } X \text{ лежит внутри } C_1 \text{ и левее } \bar{C}, \\ f_3(X), & \text{если } X \text{ лежит внутри } C_2 \text{ и правее } \bar{C}, \\ f_2(X), & \text{если } X \text{ лежит вне } C_1 \text{ и } C_2. \end{cases}$$

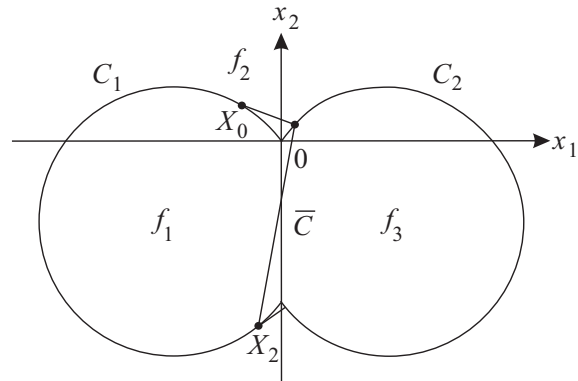


Рис. 4. Последовательность вычисленных $X_k, k \in [1 : 6]$

Выберем начальную точку $X_0 = (-1, 1)$. В [10] показано, что при использовании метода наискорейшего спуска последовательность найденных точек $X_j, j \in \mathbb{N}$, сходится к $\bar{X} = (0, 0)$. Заметим, что \bar{X} не является стационарной точкой. Из таблицы видно, что, применяя метод кодифференциального спуска, последовательность X_k (рис. 4) не застревает в \bar{X} , а сходится к точке глобального минимума $X^* = (0, -3)$ для функции $\varphi(X)$. Вычисления произведены на компьютере с процессором AMD Athlon(tm) QL-60 950 МГц, за время 0.075 секунд.

Значения X_k и $\varphi(X_k)$

k	x_1	x_2	$\varphi(X)$
0	-1.	1.	6.
1	0.193832	0.283701	1.25286
2	-0.515868	-3.63617	-1.05683
3	0.0622574	-3.09919	-2.7879
4	-0.00134449	-3.00224	-2.99552
5	6.81552e-007	-3.0000011	-2.9999977
6	3.4181991875376392e-013	-3.0000000000011551	-2.99999999997689

5. Заключение. В работе описаны основные моменты программного обеспечения для решения задач недифференцируемой оптимизации с использованием универсального точного метода вычисления кодифференциала. Созданная программа является значительным шагом перехода от теории к практическому применению квази- и кодифференциального исчисления. Ставилась цель получения точных решений оптимизационных задач, даже если от этого пострадает быстродействие. Проект был реализован на языке

программирования C++. Проведенные численные эксперименты показали эффективность предложенных методов. Планируется дальнейшее улучшение программного пакета и его приложение для решения практических задач недифференцируемой оптимизации, в частности задач математической диагностики и интервального анализа.

Выражаю благодарность рецензентам за их критику, благодаря которой статья приняла свой окончательный вид.

СПИСОК ЛИТЕРАТУРЫ

1. Демьянов В.Ф., Рубинов А.М. Основы негладкого анализа и квазидифференциального исчисления. М.: Наука, 1990.
2. Демьянов В.Ф., Васильев Л.В. Недифференцируемая оптимизация. М.: Наука, 1981.
3. Minchenko L., Volosevich A. Strongly differentiable multifunctions and directional differentiability of marginal functions // Quasidifferentiability and Related Topics. Dordrecht: Kluwer Academic Publishers, 2000. 163–171.
4. Андромонов М.Ю., Тамасян Г.Ш. Реализация аналитического кодифференцирования в пакете MatLab // Вычислительные методы и программирование. 2007. 8. 1–5.
5. Burks A.W., Warren D.W., Wright J.B. An analysis of a logical machine using parenthesis-free notation // Mathematical Tables and Other Aids to Computation. 1954. 8, N 46. 53–57.
6. Pogorzelski H.A. Reviewed work(s): remarks on Nicod's axiom and on "Generalizing Deduction" by Jan Lukasiewicz; Jerzy Slupecki; Panstwowe Wydawnictwo Naukowe // The Journal of Symbolic Logic. 1965. 30, N 3. 376–377.
7. Препарата Ф., Шеймос М. Вычислительная геометрия: Введение. М.: Мир, 1989.
8. Kallay M. Convex hull algorithms in higher dimensions. Unpublished manuscript. Univ. of Oklahoma, Dept. of Mathematics. Norman, 1981.
9. Barber C.B., Dobkin D.P., Huhdanpaa H.T. The Quickhull algorithm for convex hulls // ACM Trans. on Mathematical Software. 1996. 22. 469–483 (<http://www.qhull.org>).
10. Демьянов В.Ф., Малоземов В.Н. Введение в минимакс. М.: Наука, 1972.
11. Zarantonello E.H. Projection on convex sets in Hilbert space and spectral theory // Contributions to Nonlinear Functional Analysis / Ed. by E. Zarantonello. New York: Academic Press, 1971. 237–424.
12. Wolfe P.H. Finding the nearest point in a polytope // Math. Programm. 1976. 11, N 2. 128–149.
13. Gartner B., Jaggi M. Corsets for polytope distance // Proc. of the 25th Annual Symposium on Computational Geometry (SCG-09). Vol. 33. New York: ACM Press, 2009.
14. Карманов В.Г. Математическое программирование. М.: Физматлит, 2004.
15. Сукач М.П. Нахождение минимального расстояния от точки до выпуклого многогранника // Процессы управления и устойчивость. Труды 41-й Международной научной конференции аспирантов и студентов / Под ред. Н.В. Смирнова и Г.Ш. Тамасяна. СПб.: Издат. Дом С.-Петерб. гос. ун-та, 2010. 53–58.

Поступила в редакцию
19.05.2012
