

Министерство образования и науки Российской Федерации  
федеральное государственное бюджетное образовательное  
учреждение высшего профессионального образования  
«Сибирский государственный аэрокосмический университет  
имени академика М.Ф. Решетнева»

Сергиенко Антон Борисович

**Тестовые функции для глобальной оптимизации. v.1.20**

Красноярск – 2013

# Оглавление

<b>Условные обозначения</b>	<b>8</b>
<b>Введение</b>	<b>9</b>
<b>1 Задачи вещественной оптимизации</b>	<b>10</b>
1.1 Функция Ackley . . . . .	10
1.1.1 Описание функции . . . . .	10
1.1.2 Параметры для алгоритмов оптимизации . . . . .	11
1.1.3 Основная задача и подзадачи . . . . .	11
1.1.4 Нахождение ошибки оптимизации . . . . .	12
1.1.5 Свойства задачи . . . . .	12
1.1.6 Реализация . . . . .	12
1.1.7 Ссылки . . . . .	13
1.2 Функция Гипер-эллипсоид . . . . .	13
1.2.1 Описание функции . . . . .	13
1.2.2 Параметры для алгоритмов оптимизации . . . . .	14
1.2.3 Основная задача и подзадачи . . . . .	15
1.2.4 Нахождение ошибки оптимизации . . . . .	15
1.2.5 Свойства задачи . . . . .	16
1.2.6 Реализация . . . . .	16
1.2.7 Ссылки . . . . .	16
1.3 Эллиптический параболоид . . . . .	17
1.3.1 Описание функции . . . . .	17
1.3.2 Параметры для алгоритмов оптимизации . . . . .	18
1.3.3 Основная задача и подзадачи . . . . .	18
1.3.4 Нахождение ошибки оптимизации . . . . .	18

1.3.5	Свойства задачи . . . . .	19
1.3.6	Реализация . . . . .	19
1.3.7	Ссылки . . . . .	20
1.4	Функция Растригина . . . . .	20
1.4.1	Описание функции . . . . .	20
1.4.2	Параметры для алгоритмов оптимизации . . . . .	21
1.4.3	Основная задача и подзадачи . . . . .	21
1.4.4	Нахождение ошибки оптимизации . . . . .	21
1.4.5	Свойства задачи . . . . .	22
1.4.6	Реализация . . . . .	22
1.4.7	Ссылки . . . . .	23
1.5	Функция Розенброка . . . . .	23
1.5.1	Описание функции . . . . .	23
1.5.2	Параметры для алгоритмов оптимизации . . . . .	24
1.5.3	Основная задача и подзадачи . . . . .	24
1.5.4	Нахождение ошибки оптимизации . . . . .	25
1.5.5	Свойства задачи . . . . .	25
1.5.6	Реализация . . . . .	25
1.5.7	Ссылки . . . . .	26
1.6	Функция Развернутый гипер-эллипсоид . . . . .	26
1.6.1	Описание функции . . . . .	26
1.6.2	Параметры для алгоритмов оптимизации . . . . .	27
1.6.3	Основная задача и подзадачи . . . . .	27
1.6.4	Нахождение ошибки оптимизации . . . . .	28
1.6.5	Свойства задачи . . . . .	28
1.6.6	Реализация . . . . .	28
1.6.7	Ссылки . . . . .	29
1.7	Функция Step (модифицированная версия De Jong 3) . . . . .	29
1.7.1	Описание функции . . . . .	29
1.7.2	Параметры для алгоритмов оптимизации . . . . .	31
1.7.3	Основная задача и подзадачи . . . . .	31
1.7.4	Нахождение ошибки оптимизации . . . . .	31

1.7.5	Свойства задачи . . . . .	32
1.7.6	Реализация . . . . .	32
1.7.7	Ссылки . . . . .	33
1.8	Аддитивная потенциальная функция . . . . .	33
1.8.1	Описание функции . . . . .	33
1.8.2	Параметры для алгоритмов оптимизации . . . . .	34
1.8.3	Основная задача и подзадачи . . . . .	34
1.8.4	Нахождение ошибки оптимизации . . . . .	34
1.8.5	Свойства задачи . . . . .	35
1.8.6	Реализация . . . . .	35
1.8.7	Ссылки . . . . .	36
1.9	Функция Химмельблау . . . . .	36
1.9.1	Описание функции . . . . .	36
1.9.2	Параметры для алгоритмов оптимизации . . . . .	37
1.9.3	Основная задача и подзадачи . . . . .	37
1.9.4	Нахождение ошибки оптимизации . . . . .	37
1.9.5	Свойства задачи . . . . .	38
1.9.6	Реализация . . . . .	38
1.9.7	Ссылки . . . . .	39
1.10	Функция Катникова . . . . .	39
1.10.1	Описание функции . . . . .	39
1.10.2	Параметры для алгоритмов оптимизации . . . . .	40
1.10.3	Основная задача и подзадачи . . . . .	40
1.10.4	Нахождение ошибки оптимизации . . . . .	40
1.10.5	Свойства задачи . . . . .	41
1.10.6	Реализация . . . . .	41
1.10.7	Ссылки . . . . .	42
1.11	Функция Multiextremal3 . . . . .	42
1.11.1	Описание функции . . . . .	42
1.11.2	Параметры для алгоритмов оптимизации . . . . .	43
1.11.3	Основная задача и подзадачи . . . . .	43
1.11.4	Нахождение ошибки оптимизации . . . . .	43

1.11.5 Свойства задачи . . . . .	44
1.11.6 Реализация . . . . .	44
1.11.7 Ссылки . . . . .	45
1.12 Функция Multiextremal4 . . . . .	45
1.12.1 Описание функции . . . . .	45
1.12.2 Параметры для алгоритмов оптимизации . . . . .	46
1.12.3 Основная задача и подзадачи . . . . .	46
1.12.4 Нахождение ошибки оптимизации . . . . .	46
1.12.5 Свойства задачи . . . . .	47
1.12.6 Реализация . . . . .	47
1.12.7 Ссылки . . . . .	48
1.13 Мультипликативная потенциальная функция . . . . .	48
1.13.1 Описание функции . . . . .	48
1.13.2 Параметры для алгоритмов оптимизации . . . . .	49
1.13.3 Основная задача и подзадачи . . . . .	49
1.13.4 Нахождение ошибки оптимизации . . . . .	49
1.13.5 Свойства задачи . . . . .	50
1.13.6 Реализация . . . . .	50
1.13.7 Ссылки . . . . .	51
1.14 Функция Rana . . . . .	51
1.14.1 Описание функции . . . . .	51
1.14.2 Параметры для алгоритмов оптимизации . . . . .	52
1.14.3 Основная задача и подзадачи . . . . .	52
1.14.4 Нахождение ошибки оптимизации . . . . .	52
1.14.5 Свойства задачи . . . . .	53
1.14.6 Реализация . . . . .	53
1.14.7 Ссылки . . . . .	54
1.15 Функция ReverseGriewank . . . . .	54
1.15.1 Описание функции . . . . .	54
1.15.2 Параметры для алгоритмов оптимизации . . . . .	55
1.15.3 Основная задача и подзадачи . . . . .	55
1.15.4 Нахождение ошибки оптимизации . . . . .	55

1.15.5 Свойства задачи . . . . .	56
1.15.6 Реализация . . . . .	56
1.15.7 Ссылки . . . . .	57
1.16 Функция Сомбреро . . . . .	57
1.16.1 Описание функции . . . . .	57
1.16.2 Параметры для алгоритмов оптимизации . . . . .	58
1.16.3 Основная задача и подзадачи . . . . .	58
1.16.4 Нахождение ошибки оптимизации . . . . .	58
1.16.5 Свойства задачи . . . . .	59
1.16.6 Реализация . . . . .	59
1.16.7 Ссылки . . . . .	60
1.17 Функция Multiextremal . . . . .	60
1.17.1 Описание функции . . . . .	60
1.17.2 Параметры для алгоритмов оптимизации . . . . .	61
1.17.3 Основная задача и подзадачи . . . . .	61
1.17.4 Нахождение ошибки оптимизации . . . . .	62
1.17.5 Свойства задачи . . . . .	62
1.17.6 Реализация . . . . .	62
1.17.7 Ссылки . . . . .	63
1.18 Функция Multiextremal2 . . . . .	63
1.18.1 Описание функции . . . . .	63
1.18.2 Параметры для алгоритмов оптимизации . . . . .	64
1.18.3 Основная задача и подзадачи . . . . .	64
1.18.4 Нахождение ошибки оптимизации . . . . .	65
1.18.5 Свойства задачи . . . . .	65
1.18.6 Реализация . . . . .	65
1.18.7 Ссылки . . . . .	66
1.19 Функция волна . . . . .	66
1.19.1 Описание функции . . . . .	66
1.19.2 Параметры для алгоритмов оптимизации . . . . .	67
1.19.3 Основная задача и подзадачи . . . . .	67
1.19.4 Нахождение ошибки оптимизации . . . . .	68

1.19.5 Свойства задачи . . . . .	68
1.19.6 Реализация . . . . .	68
1.19.7 Ссылки . . . . .	69
<b>2 Задачи бинарной оптимизации</b>	<b>70</b>
2.1 Сумма всех элементов бинарного вектора . . . . .	70
2.1.1 Описание функции . . . . .	70
2.1.2 Основная задача и подзадачи . . . . .	71
2.1.3 Нахождение ошибки оптимизации . . . . .	71
2.1.4 Свойства задачи . . . . .	72
2.1.5 Реализация . . . . .	72
<b>Литература</b>	<b>73</b>

# Условные обозначения

$a \in A$  — элемент  $a$  принадлежит множеству  $A$ .

$\bar{x}$  — обозначение вектора.

$\arg f(x)$  — возвращает аргумент  $x$ , при котором функция принимает значение  $f(x)$ .

$\text{Random}(X)$  — случайный выбор элемента из множества  $X$  с равной вероятностью.

$\text{Random}(\{x^i \mid p^i\})$  — случайный выбор элемента  $x^i$  из множества  $X$ , при условии, что каждый элемент  $x^i \in X$  имеет вероятность выбора равную  $p^i$ , то есть это обозначение равнозначно предыдущему.

$\text{random}(a, b)$  — случайное действительное число из интервала  $[a; b]$ .

$\text{int}(a)$  — целая часть действительного числа  $a$ .

$\mu(X)$  — мощность множества  $X$ .

**Замечание.** Оператор присваивания обозначается через знак « $=$ », так же как и знак равенства.

**Замечание.** Индексация всех массивов в документе начинается с 1. Это стоит помнить при реализации алгоритма на С-подобных языках программирования, где индексация начинается с нуля.

**Замечание.** Вызывание трех функций:  $\text{Random}(X)$ ,  $\text{Random}(\{x_i \mid p_i\})$ ,  $\text{random}(a, b)$  — происходит каждый раз, когда по ходу выполнения формул, они встречаются. Если формула итерационная, то нельзя перед ее вызовом один раз определить, например,  $\text{random}(a, b)$  как константу и потом её использовать на протяжении всех итераций неизменной.

**Замечание.** Надстрочный индекс может обозначать как возведение в степень, так и индекс элемента. Конкретное обозначение определяется в контексте текста, в котором используется формула с надстрочным индексом.

**Замечание.** Если у нас имеется множество векторов, то подстрочный индекс обозначает номер компоненты конкретного вектора, а надстрочный индекс обозначает номер вектора во множестве, например,  $\bar{x}^i \in X$  ( $i = \overline{1, N}$ ),  $\bar{x}_j^i \in \{0; 1\}$ , ( $j = \overline{1, n}$ ). В случае, если вектор имеет свое обозначение в виде подстрочной надписи, то компоненты вектора проставляются за скобками, например,  $(\bar{x}_{\max})_j = 0$  ( $j = \overline{1, n}$ ).

**Замечание.** При выводе матриц и векторов элементы могут разделяться как пробелом, так и точкой с запятой, то есть обе записи  $(1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1)^T$  и  $(1; 1; 1; 1; 1; 1; 1; 1)^T$  допустимы.

**Замечание.** При выводе множеств элементы разделяются только точкой с запятой, то есть допустима только такая запись:  $\{1; 1; 1; 1; 1; 1; 1\}^T$ .

# Введение

В данном документе рассмотрено множество тестовых функций, которые можно использовать для проведения исследований алгоритмов оптимизации. К каждой функции дано подробное описание, график (если это возможно), свойства и параметров, которые позволяют единственно проводить сравнения разных алгоритмов оптимизации во избежания несостыковок с точки зрения разного понимания нахождения ошибки, точности работы алгоритмом.

Данный документ представляет его версию **1.20** от 29 декабря 2013 г.

Последнюю версию документа можно найти по адресу:

<https://github.com/Harrix/HarrixTestFunctions>

Там же можно найти реализацию тестовых функций в среде Mathcad.

Тестовые функции реализованы на языке C++ в библиотеке **HarrixMathLibrary** в разделе «Тестовые функции для оптимизации», которую можно найти по адресу:

<https://github.com/Harrix/HarrixMathLibrary>.

Все библиографические материалы, которые используются в документе, приведены в виде скриншотов, скринов, документов в папке **\_Biblio** на <https://github.com/Harrix/HarrixTestFunctions>.

С автором можно связаться по адресу [sergienkoanton@mail.ru](mailto:sergienkoanton@mail.ru) или <http://vk.com/harrix>.

Сайт автора, где публикуются последние новости: <http://blog.harrix.org/>, а проекты расположаются по адресу <http://harrix.org/>.

# Глава 1

## Задачи вещественной оптимизации

### 1.1 Функция Ackley

#### 1.1.1 Описание функции

**Идентификатор:**

MHL\_TestFunction\_Ackley.

**Наименование:**

Функция Ackley.

**Тип:**

Задача вещественной оптимизации.

**Формула** (целевая функция):

$$f(\bar{x}) = 20 + e - 20e^{-0.2\sqrt{\frac{1}{n} \sum_{i=1}^n \bar{x}_i^2}} - e^{\frac{1}{n} \sum_{i=1}^n \cos(2\pi \cdot \bar{x}_i)}, \text{ где } \quad (1.1)$$

$\bar{x} \in X$ ,  $\bar{x}_j \in [Left_j; Right_j]$ ,  $Left_j = -5$ ,  $Right_j = 5$ ,  $j = \overline{1, n}$ .

**Обозначение:**

$\bar{x}$  — вещественный вектор;

$n$  — размерность вещественного вектора.

**Решаемая задача оптимизации:**

$\bar{x}_{min} = \arg \min_{\bar{x} \in X} f(\bar{x})$ .

**Точка минимума:**

$\bar{x}_{min} = (0, 0, \dots, 0)^T$ , то есть  $(\bar{x}_{min})_j = 0$  ( $j = \overline{1, n}$ ).

**Минимум функции:**

$f(\bar{x}_{min}) = 0$ .

**График:**

Рисунок 1.1 на с. 11 стр.

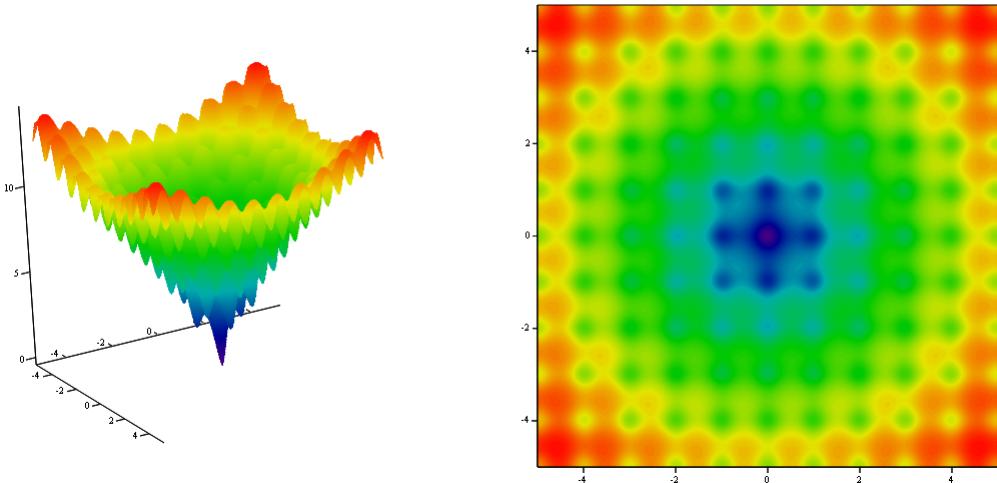


Рисунок 1.1. Функция Ackley

### 1.1.2 Параметры для алгоритмов оптимизации

**Точность вычислений:**

$$\varepsilon = 0.025.$$

**Число интервалов, на которые предполагается разбивать каждую компоненту вектора  $\bar{x}$  в пределах своего изменения** (для алгоритмов дискретной оптимизации) :

$$NumberOfParts_j = 4095 \ (j = \overline{1, n}).$$

**Для этого длина бинарной строки для  $x_j$  координаты равна** (для алгоритмов бинарной оптимизации) :

$$(k_2)_j = 12 \ (j = \overline{1, n}).$$

**Замечание:**  $NumberOfParts_j$  выбирается как минимальное число, удовлетворяющее соотношению:

$$NumberOfParts_j = 2^{(k_2)_j} - 1 \geq \frac{10 (Right_j - Left_j)}{\varepsilon}, \text{ где } (k_2)_j \in \mathbb{N}, (j = \overline{1, n}).$$

### 1.1.3 Основная задача и подзадачи

**Изменяемый параметр:**

$n$  — размерность вещественного вектора.

**Значение в основной задаче:**

$$n = 2.$$

**Подзадача №2:**

$$n = 3.$$

**Подзадача №3:**

$$n = 4.$$

**Подзадача №4:**

$$n = 5.$$

**Подзадача №5:**

$$n = 10.$$

**Подзадача №6:**

$$n = 20.$$

**Подзадача №7:**

$$n = 30.$$

#### 1.1.4 Нахождение ошибки оптимизации

Пусть в результате работы алгоритма оптимизации за  $N$  запусков мы нашли решения  $\bar{x}_{submin}^k$  со значениями целевой функции  $f(\bar{x}_{submin}^k)$  соответственно ( $k = \overline{1, N}$ ). Используем три вида ошибок:

**Надёжность:**

$$R = \frac{\sum_{k=1}^N S(\bar{x}_{submin}^k)}{N}, \text{ где}$$
$$S(\bar{x}_{submin}^k) = \begin{cases} 1, & \text{если } |(\bar{x}_{submin}^k)_j - (\bar{x}_{min})_j| < \varepsilon, j = \overline{1, n}; \\ 0, & \text{иначе.} \end{cases}$$

**Ошибка по входным параметрам:**

$$E_x = \frac{\sum_{k=1}^N \left( \frac{\sqrt{\sum_{j=1}^n ((\bar{x}_{submin}^k)_j - (\bar{x}_{min})_j)^2}}{n} \right)}{N}.$$

**Ошибка по значениям целевой функции:**

$$E_f = \frac{\sum_{k=1}^N |f(\bar{x}_{submin}^k) - f(\bar{x}_{min})|}{N}.$$

#### 1.1.5 Свойства задачи

**Условной или безусловной оптимизации:** Задача безусловной оптимизации.

**Одномерной или многомерной оптимизации:** Многомерной:  $n$ .

**Функция унимодальная или многоэкстремальная:** Функция многоэкстремальная.

**Функция стохастическая или нет:** Функция не стохастическая.

**Особенности:** Нет.

#### 1.1.6 Реализация

Реализация функции взята из библиотеки HarrixMathLibrary в разделе «Тестовые функции для оптимизации», которую можно найти по адресу <https://github.com/Harrix/HarrixMathLibrary>.

Код 1.1. Код функции MHL\_TestFunction\_Ackley

```
double MHL_TestFunction_Ackley(double *x, int VMHL_N)
{
    /*
```

```

Функция многих переменных: Ackley.
Тестовая функция вещественной оптимизации.
Входные параметры:
x - указатель на исходный массив;
VMHL_N - размер массива x.
Возвращаемое значение:
Значение тестовой функции в точке x.
*/
double VMHL_Result;
double f1,f2=0;
f1=exp(-0.2*sqrt(TMHL_SumSquareVector(x,VMHL_N)/double(VMHL_N)));
for (int i=0;i<VMHL_N;i++) f2=f2+cos(2.*MHL_PI*x[i]);
f2=exp(f2/double(VMHL_N));
VMHL_Result=20.+exp(1)-20.*f1-f2;
return VMHL_Result;
}

```

### 1.1.7 Ссылки

Данная функция приводится в следующих источниках:

1. [1, стр. 5] — Empirical review of standard benchmark functions using evolutionary global optimization.
2. [2] — Ackley's Function.

## 1.2 Функция Гипер-эллипсоид

### 1.2.1 Описание функции

**Идентификатор:** MHL\_TestFunction\_HyperEllipsoid.

**Наименование:** Гипер-эллипсоид.

**Тип:** Задача вещественной оптимизации.

**Формула** (целевая функция):

$$f(\bar{x}) = \sum_{i=1}^n (i \cdot \bar{x}_i)^2, \text{ где } \quad (1.2)$$

$\bar{x} \in X, \bar{x}_j \in [Left_j; Right_j], Left_j = -5, Right_j = 5, j = \overline{1, n}.$

**Обозначение:**

$\bar{x}$  — вещественный вектор;

$n$  — размерность вещественного вектора.

**Решаемая задача оптимизации:**

$$\bar{x}_{min} = \arg \min_{\bar{x} \in X} f(\bar{x}).$$

**Точка минимума:**

$$\bar{x}_{min} = (0, 0, \dots, 0)^T, \text{ то есть } (\bar{x}_{min})_j = 0 \ (j = \overline{1, n}).$$

**Минимум функции:**

$$f(\bar{x}_{min}) = 0.$$

**График:**

Рисунок 1.2 на с. 14 стр.

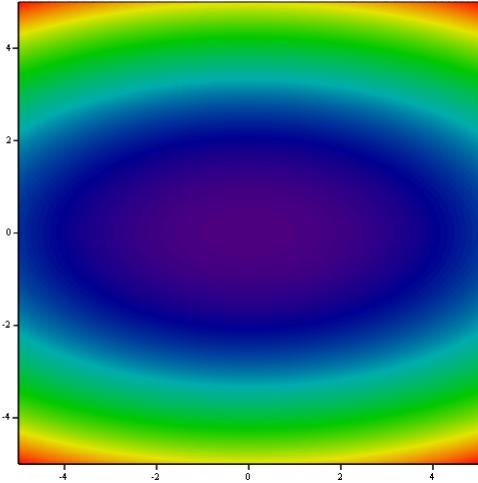
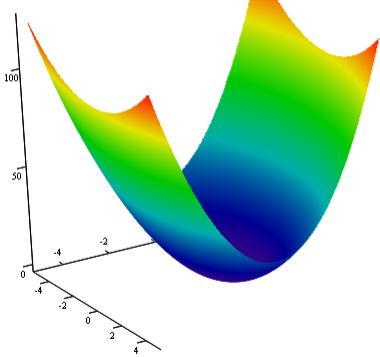


Рисунок 1.2. Гипер-эллипсоид

## 1.2.2 Параметры для алгоритмов оптимизации

**Точность вычислений:**

$$\varepsilon = 0.025.$$

**Число интервалов, на которые предполагается разбивать каждую компоненту вектора  $\bar{x}$  в пределах своего изменения** (для алгоритмов дискретной оптимизации) :

$$NumberOfParts_j = 4095 \ (j = \overline{1, n}).$$

**Для этого длина бинарной строки для  $x_j$  координаты равна** (для алгоритмов бинарной оптимизации) :

$$(k_2)_j = 12 \ (j = \overline{1, n}).$$

**Замечание:**  $NumberOfParts_j$  выбирается как минимальное число, удовлетворяющее соотношению:

$$NumberOfParts_j = 2^{(k_2)_j} - 1 \geq \frac{10 (Right_j - Left_j)}{\varepsilon}, \text{ где } (k_2)_j \in \mathbb{N}, (j = \overline{1, n}).$$

### 1.2.3 Основная задача и подзадачи

<b>Изменяемый параметр:</b>	$n$ — размерность вещественного вектора.
<b>Значение в основной задаче:</b>	$n = 2$ .
<b>Подзадача №2:</b>	$n = 3$ .
<b>Подзадача №3:</b>	$n = 4$ .
<b>Подзадача №4:</b>	$n = 5$ .
<b>Подзадача №5:</b>	$n = 10$ .
<b>Подзадача №6:</b>	$n = 20$ .
<b>Подзадача №7:</b>	$n = 30$ .

### 1.2.4 Нахождение ошибки оптимизации

Пусть в результате работы алгоритма оптимизации за  $N$  запусков мы нашли решения  $\bar{x}_{submin}^k$  со значениями целевой функции  $f(\bar{x}_{submin}^k)$  соответственно ( $k = \overline{1, N}$ ). Используем три вида ошибок:

**Надёжность:**

$$R = \frac{\sum_{k=1}^N S(\bar{x}_{submin}^k)}{N}, \text{ где}$$

$$S(\bar{x}_{submin}^k) = \begin{cases} 1, & \text{если } \left| (\bar{x}_{submin}^k)_j - (\bar{x}_{min})_j \right| < \varepsilon, j = \overline{1, n}; \\ 0, & \text{иначе.} \end{cases}$$

**Ошибка по входным параметрам:**

$$E_x = \frac{\sum_{k=1}^N \left( \sqrt{\sum_{j=1}^n ((\bar{x}_{submin}^k)_j - (\bar{x}_{min})_j)^2} \right)}{N}.$$

**Ошибка по значениям целевой функции:**

$$E_f = \frac{\sum_{k=1}^N |f(\bar{x}_{submin}^k) - f(\bar{x}_{min})|}{N}.$$

## 1.2.5 Свойства задачи

**Условной или безусловной оптимизации:** Задача безусловной оптимизации.

**Одномерной или многомерной оптимизации:** Многомерной:  $n$ .

**Функция унимодальная или многоэкстремальная:** Функция унимодальная.

**Функция стохастическая или нет:** Функция не стохастическая.

**Особенности:** Нет.

## 1.2.6 Реализация

Реализация функции взята из библиотеки HarrixMathLibrary в разделе «Тестовые функции для оптимизации», которую можно найти по адресу <https://github.com/Harrix/HarrixMathLibrary>.

Код 1.2. Код функции MHL\_TestFunction\_HyperEllipsoid

```
double MHL_TestFunction_HyperEllipsoid(double *x, int VMHL_N)
{
/*
Функция многих переменных: Гипер-эллипсоид.
Тестовая функция вещественной оптимизации.
Входные параметры:
    x - указатель на исходный массив;
    VMHL_N - размер массива x.
Возвращаемое значение:
    Значение тестовой функции в точке x.
*/
double VMHL_Result=0;

for (int i=0;i<VMHL_N;i++)
    VMHL_Result += (i+1)*(i+1)*x[i]*x[i];

return VMHL_Result;
}
```

## 1.2.7 Ссылки

В данном виде тестовую функцию в литературе не нашел. Обычно используется несколько иной вид этой функции, когда  $i$  не возводится в квадрат.

## 1.3 Эллиптический параболоид

### 1.3.1 Описание функции

**Идентификатор:**

MHL\_TestFunction\_ParaboloidOfRevolution.

**Наименование:**

Эллиптический параболоид.

**Тип:**

Задача вещественной оптимизации.

**Формула** (целевая функция):

$$f(\bar{x}) = \sum_{i=1}^n \bar{x}_i^2, \text{ где} \quad (1.3)$$

$\bar{x} \in X, \bar{x}_j \in [Left_j; Right_j], Left_j = -2, Right_j = 2, j = \overline{1, n}.$

**Обозначение:**

$\bar{x}$  — вещественный вектор;

$n$  — размерность вещественного вектора.

**Решаемая задача оптимизации:**

$\bar{x}_{min} = \arg \min_{\bar{x} \in X} f(\bar{x}).$

**Точка минимума:**

$\bar{x}_{min} = (0, 0, \dots, 0)^T$ , то есть  $(\bar{x}_{min})_j = 0 (j = \overline{1, n}).$

**Минимум функции:**

$f(\bar{x}_{min}) = 0.$

**График:**

Рисунок 1.3 на с. 17

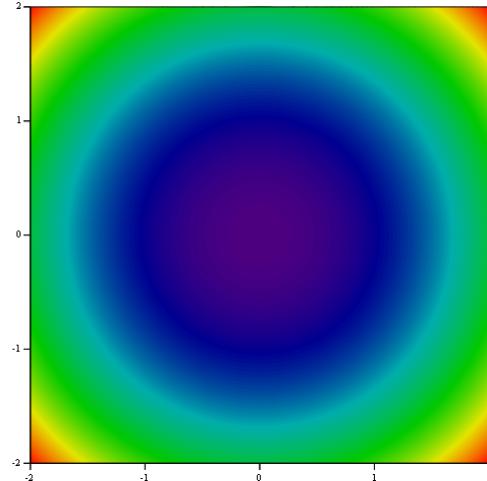
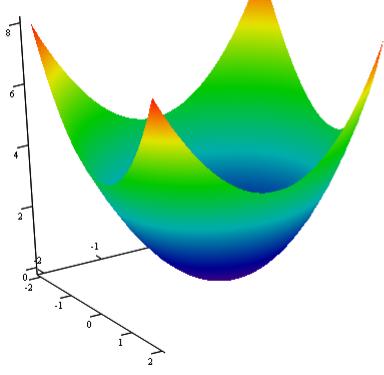


Рисунок 1.3. Эллиптический параболоид

### 1.3.2 Параметры для алгоритмов оптимизации

**Точность вычислений:**

$$\varepsilon = 0.01.$$

**Число интервалов, на которые предполагается разбивать каждую компоненту вектора  $\bar{x}$  в пределах своего изменения** (для алгоритмов дискретной оптимизации) :

**Для этого длина бинарной строки для  $x_j$  координаты равна** (для алгоритмов бинарной оптимизации) :

**Замечание:**  $NumberOfParts_j$  выбирается как минимальное число, удовлетворяющее соотношению:

$$NumberOfParts_j = 2^{(k_2)_j} - 1 \geq \frac{10 (Right_j - Left_j)}{\varepsilon}, \text{ где } (k_2)_j \in \mathbb{N}, (j = \overline{1, n}).$$

### 1.3.3 Основная задача и подзадачи

**Изменяемый параметр:**

$n$  — размерность вещественного вектора.

**Значение в основной задаче:**

$$n = 2.$$

**Подзадача №2:**

$$n = 3.$$

**Подзадача №3:**

$$n = 4.$$

**Подзадача №4:**

$$n = 5.$$

**Подзадача №5:**

$$n = 10.$$

**Подзадача №6:**

$$n = 20.$$

**Подзадача №7:**

$$n = 30.$$

### 1.3.4 Нахождение ошибки оптимизации

Пусть в результате работы алгоритма оптимизации за  $N$  запусков мы нашли решения  $\bar{x}_{submin}^k$  со значениями целевой функции  $f(\bar{x}_{submin}^k)$  соответственно ( $k = \overline{1, N}$ ). Используем три вида ошибок:

**Надёжность:**

$$R = \frac{\sum_{k=1}^N S(\bar{x}_{submin}^k)}{N}, \text{ где}$$

$$S(\bar{x}_{submin}^k) = \begin{cases} 1, & \text{если } \left| (\bar{x}_{submin}^k)_j - (\bar{x}_{min})_j \right| < \varepsilon, j = \overline{1, n}; \\ 0, & \text{иначе.} \end{cases}$$

**Ошибка по входным параметрам:**

$$E_x = \frac{\sum_{k=1}^N \left( \frac{\sqrt{\sum_{j=1}^n ((\bar{x}_{submin}^k)_j - (\bar{x}_{min})_j)^2}}{n} \right)}{N}.$$

**Ошибка по значениям целевой функции:**

$$E_f = \frac{\sum_{k=1}^N |f(\bar{x}_{submin}^k) - f(\bar{x}_{min})|}{N}.$$

### 1.3.5 Свойства задачи

**Условной или безусловной оптимизации:** Задача безусловной оптимизации.

**Одномерной или многомерной оптимизации:** Многомерной:  $n$ .

**Функция унимодальная или многоэкстремальная:** Функция унимодальная.

**Функция стохастическая или нет:** Функция не стохастическая.

**Особенности:** Нет.

### 1.3.6 Реализация

Реализация функции взята из библиотеки HarrixMathLibrary в разделе «Тестовые функции для оптимизации», которую можно найти по адресу <https://github.com/Harrix/HarrixMathLibrary>.

Код 1.3. Код функции MHL\_TestFunction\_ParaboloidOfRevolution

```
double MHL_TestFunction_ParaboloidOfRevolution(double *x, int VMHL_N)
{
/*
Функция многих переменных: Эллиптический параболоид.
Тестовая функция вещественной оптимизации.
Входные параметры:
    x - указатель на исходный массив;
    VMHL_N - размер массива x.
Возвращаемое значение:
    Значение тестовой функции в точке x.
*/
double VMHL_Result=0;
for (int i=0;i<VMHL_N;i++) VMHL_Result+=x[i]*x[i];
return VMHL_Result;
}
```

### 1.3.7 Ссылки

Данная функция приводится в следующих источниках:

1. [3] — Paraboloid.

## 1.4 Функция Растигина

### 1.4.1 Описание функции

**Идентификатор:**

MHL\_TestFunction\_Rastrigin.

**Наименование:**

Функция Растигина.

**Тип:**

Задача вещественной оптимизации.

**Формула** (целевая функция):

$$f(\bar{x}) = 10n + \sum_{i=1}^n (\bar{x}_i^2 - 10 \cdot \cos(2\pi \cdot \bar{x}_i)), \text{ где} \quad (1.4)$$

$\bar{x} \in X, \bar{x}_j \in [Left_j; Right_j], Left_j = -5, Right_j = 5, j = \overline{1, n}.$

**Обозначение:**

$\bar{x}$  — вещественный вектор;

$n$  — размерность вещественного вектора.

**Решаемая задача оптимизации:**

$\bar{x}_{min} = \arg \min_{\bar{x} \in X} f(\bar{x}).$

**Точка минимума:**

$\bar{x}_{min} = (0, 0, \dots, 0)^T$ , то есть  $(\bar{x}_{min})_j = 0 (j = \overline{1, n}).$

**Минимум функции:**

$f(\bar{x}_{min}) = 0.$

**График:**

Рисунок 1.4 на 20 стр.

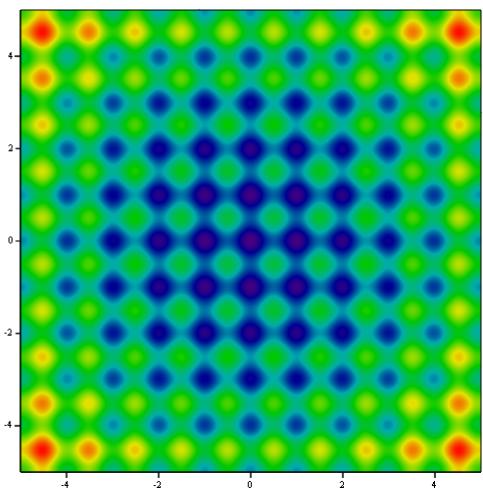
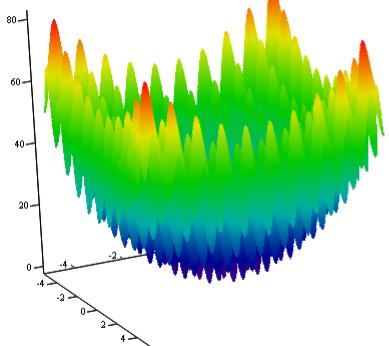


Рисунок 1.4. Функция Растигина

## 1.4.2 Параметры для алгоритмов оптимизации

**Точность вычислений:**  $\varepsilon = 0.025$ .

**Число интервалов, на которые предполагается разбивать каждую компоненту вектора  $\bar{x}$  в пределах своего изменения** (для алгоритмов дискретной оптимизации) :

**Для этого длина бинарной строки для  $x_j$  координаты равна** (для алгоритмов бинарной оптимизации) :

**Замечание:**  $NumberOfParts_j$  выбирается как минимальное число, удовлетворяющее соотношению:

$$NumberOfParts_j = 2^{(k_2)_j} - 1 \geq \frac{10 (Right_j - Left_j)}{\varepsilon}, \text{ где } (k_2)_j \in \mathbb{N}, (j = \overline{1, n}).$$

## 1.4.3 Основная задача и подзадачи

**Изменяемый параметр:**  $n$  — размерность вещественного вектора.

**Значение в основной задаче:**  $n = 2$ .

**Подзадача №2:**  $n = 3$ .

**Подзадача №3:**  $n = 4$ .

**Подзадача №4:**  $n = 5$ .

**Подзадача №5:**  $n = 10$ .

**Подзадача №6:**  $n = 20$ .

**Подзадача №7:**  $n = 30$ .

## 1.4.4 Нахождение ошибки оптимизации

Пусть в результате работы алгоритма оптимизации за  $N$  запусков мы нашли решения  $\bar{x}_{submin}^k$  со значениями целевой функции  $f(\bar{x}_{submin}^k)$  соответственно ( $k = \overline{1, N}$ ). Используем три вида ошибок:

**Надёжность:**

$$R = \frac{\sum_{k=1}^N S(\bar{x}_{submin}^k)}{N}, \text{ где}$$

$$S(\bar{x}_{submin}^k) = \begin{cases} 1, & \text{если } \left| (\bar{x}_{submin}^k)_j - (\bar{x}_{min})_j \right| < \varepsilon, j = \overline{1, n}; \\ 0, & \text{иначе.} \end{cases}$$

**Ошибка по входным параметрам:**

$$E_x = \frac{\sum_{k=1}^N \left( \frac{\sqrt{\sum_{j=1}^n ((\bar{x}_{submin}^k)_j - (\bar{x}_{min})_j)^2}}{n} \right)}{N}.$$

**Ошибка по значениям целевой функции:**

$$E_f = \frac{\sum_{k=1}^N |f(\bar{x}_{submin}^k) - f(\bar{x}_{min})|}{N}.$$

#### 1.4.5 Свойства задачи

**Условной или безусловной оптимизации:** Задача безусловной оптимизации.

**Одномерной или многомерной оптимизации:** Многомерной:  $n$ .

**Функция унимодальная или многоэкстремальная:** Функция многоэкстремальная.

**Функция стохастическая или нет:** Функция не стохастическая.

**Особенности:** Нет.

#### 1.4.6 Реализация

Реализация функции взята из библиотеки HarrixMathLibrary в разделе «Тестовые функции для оптимизации», которую можно найти по адресу <https://github.com/Harrix/HarrixMathLibrary>.

Код 1.4. Код функции MHL\_TestFunction\_Rastrigin

```
double MHL_TestFunction_Rastrigin(double *x, int VMHL_N)
{
/*
Функция многих переменных: функция Раstrигина.
Тестовая функция вещественной оптимизации.
Входные параметры:
    x - указатель на исходный массив;
    VMHL_N - размер массива x.
Возвращаемое значение:
    Значение тестовой функции в точке x.
*/
double VMHL_Result=0;
for (int i=0;i<VMHL_N;i++) VMHL_Result+=x[i]*x[i]-10.*cos(2.*MHL_PI*x[i]);
VMHL_Result+=10*VMHL_N;
return VMHL_Result;
}
```

## 1.4.7 Ссылки

Данная функция приводится в следующих источниках:

1. [4] — Rastrigin function.
2. [5] — Non-linear Continuous Multi-Extremal Optimization.
3. [6] — Parametric Optimization.

## 1.5 Функция Розенброка

### 1.5.1 Описание функции

**Идентификатор:**

MHL\_TestFunction\_Rosenbrock.

**Наименование:**

Функция Розенброка.

**Тип:**

Задача вещественной оптимизации.

**Формула** (целевая функция):

$$f(\bar{x}) = \sum_{i=1}^{n-1} \left( 100(\bar{x}_{i+1} - \bar{x}_i^2)^2 + (1 - \bar{x}_i)^2 \right), \text{ где} \quad (1.5)$$

$\bar{x} \in X$ ,  $\bar{x}_j \in [Left_j; Right_j]$ ,  $Left_j = -2$ ,  $Right_j = 2$ ,  $j = \overline{1, n}$ .

**Обозначение:**

$\bar{x}$  — вещественный вектор;

$n$  — размерность вещественного вектора.

**Решаемая задача оптимизации:**

$\bar{x}_{min} = \arg \min_{\bar{x} \in X} f(\bar{x})$ .

**Точка минимума:**

$\bar{x}_{min} = (1, 1, \dots, 1)^T$ , то есть  $(\bar{x}_{min})_j = 1$  ( $j = \overline{1, n}$ ).

**Минимум функции:**

$f(\bar{x}_{min}) = 0$ .

**График:**

Рисунок 1.5 на с. 24 стр.

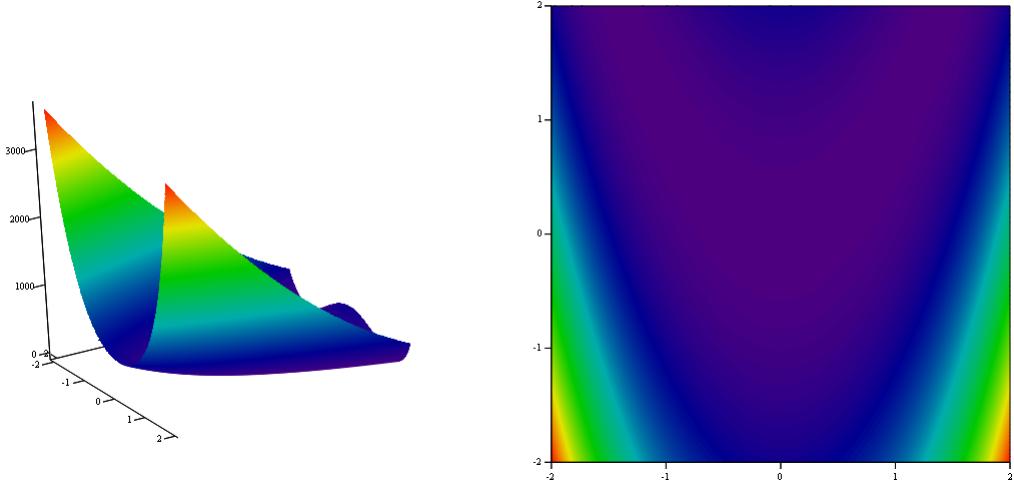


Рисунок 1.5. Функция Розенброка

### 1.5.2 Параметры для алгоритмов оптимизации

**Точность вычислений:**

$$\varepsilon = 0.01.$$

**Число интервалов, на которые предполагается разбивать каждую компоненту вектора  $\bar{x}$  в пределах своего изменения** (для алгоритмов дискретной оптимизации) :

$$NumberOfParts_j = 4095 \ (j = \overline{1, n}).$$

**Для этого длина бинарной строки для  $x_j$  координаты равна** (для алгоритмов бинарной оптимизации) :

**Замечание:**  $NumberOfParts_j$  выбирается как минимальное число, удовлетворяющее соотношению:

$$NumberOfParts_j = 2^{(k_2)_j} - 1 \geq \frac{10 (Right_j - Left_j)}{\varepsilon}, \text{ где } (k_2)_j \in \mathbb{N}, (j = \overline{1, n}).$$

### 1.5.3 Основная задача и подзадачи

**Изменяемый параметр:**

$n$  — размерность вещественного вектора.

**Значение в основной задаче:**

$$n = 2.$$

**Подзадача №2:**

$$n = 3.$$

**Подзадача №3:**

$$n = 4.$$

**Подзадача №4:**

$$n = 5.$$

**Подзадача №5:**

$$n = 10.$$

**Подзадача №6:**

$$n = 20.$$

**Подзадача №7:**

$$n = 30.$$

#### 1.5.4 Нахождение ошибки оптимизации

Пусть в результате работы алгоритма оптимизации за  $N$  запусков мы нашли решения  $\bar{x}_{submin}^k$  со значениями целевой функции  $f(\bar{x}_{submin}^k)$  соответственно ( $k = \overline{1, N}$ ). Используем три вида ошибок:

**Надёжность:**

$$R = \frac{\sum_{k=1}^N S(\bar{x}_{submin}^k)}{N}, \text{ где}$$
$$S(\bar{x}_{submin}^k) = \begin{cases} 1, & \text{если } |(\bar{x}_{submin}^k)_j - (\bar{x}_{min})_j| < \varepsilon, j = \overline{1, n}; \\ 0, & \text{иначе.} \end{cases}$$

**Ошибка по входным параметрам:**

$$E_x = \frac{\sum_{k=1}^N \left( \frac{\sqrt{\sum_{j=1}^n ((\bar{x}_{submin}^k)_j - (\bar{x}_{min})_j)^2}}{n} \right)}{N}.$$

**Ошибка по значениям целевой функции:**

$$E_f = \frac{\sum_{k=1}^N |f(\bar{x}_{submin}^k) - f(\bar{x}_{min})|}{N}.$$

#### 1.5.5 Свойства задачи

**Условной или безусловной оптимизации:** Задача безусловной оптимизации.

**Одномерной или многомерной оптимизации:** Многомерной:  $n$ .

**Функция унимодальная или многоэкстремальная:** Функция многоэкстремальная.

**Функция стохастическая или нет:** Функция не стохастическая.

**Особенности:** Нет.

#### 1.5.6 Реализация

Реализация функции взята из библиотеки HarrixMathLibrary в разделе «Тестовые функции для оптимизации», которую можно найти по адресу <https://github.com/Harrix/HarrixMathLibrary>.

Код 1.5. Код функции MHL\_TestFunction\_Rosenbrock

```
double MHL_TestFunction_Rosenbrock(double *x, int VMHL_N)
{
    /*
```

*Функция многих переменных: функция Розенброка.*

*Тестовая функция вещественной оптимизации.*

*Входные параметры:*

*x – указатель на исходный массив;*

*VMHL\_N – размер массива x.*

*Возвращаемое значение:*

*Значение тестовой функции в точке x.*

*\*/*

```
double VMHL_Result=0;
```

```
for (int i=0;i<VMHL_N-1;i++) VMHL_Result+=100.*( $x[i+1]-x[i]*x[i]$ )*( $x[i+1]-x[i]*x[i]$ )  
+(1.- $x[i]$ )*(1.- $x[i]$ );
```

```
return VMHL_Result;
```

```
}
```

## 1.5.7 Ссылки

Данная функция приводится в следующих источниках:

1. [7] – Rosenbrock function.
2. [8] – Rosenbrock Function.

## 1.6 Функция Развернутый гипер-эллипсоид

### 1.6.1 Описание функции

**Идентификатор:**

MHL\_TestFunction\_RotatedHyperEllipsoid.

**Наименование:**

Развернутый гипер-эллипсоид.

**Тип:**

Задача вещественной оптимизации.

**Формула** (целевая функция):

$$f(\bar{x}) = \sum_{i=1}^n \left( \sum_{j=1}^i \bar{x}_j \right)^2, \text{ где } \quad (1.6)$$

$\bar{x} \in X, \bar{x}_j \in [Left_j; Right_j], Left_j = -5, Right_j = 5, j = \overline{1, n}.$

**Обозначение:**

$\bar{x}$  – вещественный вектор;

$n$  – размерность вещественного вектора.

**Решаемая задача оптимизации:**

$\bar{x}_{min} = \arg \min_{\bar{x} \in X} f(\bar{x}).$

**Точка минимума:**

$\bar{x}_{min} = (0, 0, \dots, 0)^T$ , то есть  $(\bar{x}_{min})_j = 0 (j = \overline{1, n}).$

**Минимум функции:**

$f(\bar{x}_{min}) = 0.$

**График:**

Рисунок 1.6 на 27 стр.

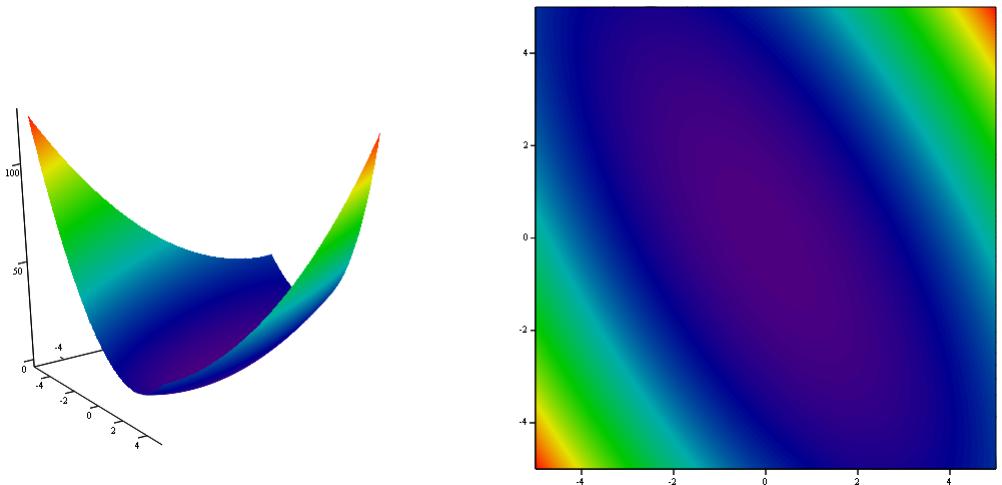


Рисунок 1.6. Развернутый гипер-эллипсоид

### 1.6.2 Параметры для алгоритмов оптимизации

**Точность вычислений:**  $\varepsilon = 0.025.$

**Число интервалов, на которые предполагается разбивать каждую компоненту вектора  $\bar{x}$  в пределах своего изменения** (для алгоритмов дискретной оптимизации) :

**Для этого длина бинарной строки для  $x_j$  координаты равна** (для алгоритмов бинарной оптимизации) :  $(k_2)_j = 12 (j = \overline{1, n}).$

**Замечание:**  $NumberOfParts_j$  выбирается как минимальное число, удовлетворяющее соотношению:

$$NumberOfParts_j = 2^{(k_2)_j} - 1 \geq \frac{10 (Right_j - Left_j)}{\varepsilon}, \text{ где } (k_2)_j \in \mathbb{N}, (j = \overline{1, n}).$$

### 1.6.3 Основная задача и подзадачи

**Изменяемый параметр:**  $n$  — размерность вещественного вектора.

**Значение в основной задаче:**  $n = 2.$

**Подзадача №2:**  $n = 3.$

**Подзадача №3:**  $n = 4.$

**Подзадача №4:**  $n = 5.$

**Подзадача №5:**  $n = 10.$

**Подзадача №6:**  $n = 20.$

**Подзадача №7:**  $n = 30.$

## 1.6.4 Нахождение ошибки оптимизации

Пусть в результате работы алгоритма оптимизации за  $N$  запусков мы нашли решения  $\bar{x}_{submin}^k$  со значениями целевой функции  $f(\bar{x}_{submin}^k)$  соответственно ( $k = \overline{1, N}$ ). Используем три вида ошибок:

**Надёжность:**

$$R = \frac{\sum_{k=1}^N S(\bar{x}_{submin}^k)}{N}, \text{ где}$$
$$S(\bar{x}_{submin}^k) = \begin{cases} 1, & \text{если } |(\bar{x}_{submin}^k)_j - (\bar{x}_{min})_j| < \varepsilon, j = \overline{1, n}; \\ 0, & \text{иначе.} \end{cases}$$

**Ошибка по входным параметрам:**

$$E_x = \frac{\sum_{k=1}^N \left( \frac{\sqrt{\sum_{j=1}^n ((\bar{x}_{submin}^k)_j - (\bar{x}_{min})_j)^2}}{n} \right)}{N}.$$

**Ошибка по значениям целевой функции:**

$$E_f = \frac{\sum_{k=1}^N |f(\bar{x}_{submin}^k) - f(\bar{x}_{min})|}{N}.$$

## 1.6.5 Свойства задачи

**Условной или безусловной оптимизации:** Задача безусловной оптимизации.

**Одномерной или многомерной оптимизации:** Многомерной:  $n$ .

**Функция унимодальная или многоэкстремальная:** Функция унимодальная.

**Функция стохастическая или нет:** Функция не стохастическая.

**Особенности:** Нет.

## 1.6.6 Реализация

Реализация функции взята из библиотеки HarrixMathLibrary в разделе «Тестовые функции для оптимизации», которую можно найти по адресу <https://github.com/Harrix/HarrixMathLibrary>.

Код 1.6. Код функции MHL\_TestFunction\_RotatedHyperEllipsoid

```
double MHL_TestFunction_RotatedHyperEllipsoid(double *x, int VMHL_N)
{
    /*
```

*Функция многих переменных: Развёрнутый гипер-эллипсоид.*

*Тестовая функция вещественной оптимизации.*

*Входные параметры:*

*x – указатель на исходный массив;*

*VMHL\_N – размер массива x.*

*Возвращаемое значение:*

*Значение тестовой функции в точке x.*

*\*/*

**double** VMHL\_Result=0;

**double** f;

**for** (**int** i=0;i<VMHL\_N;i++)

{

f=0;

**for** (**int** j=0;j<i+1;j++)

f += x[j];

VMHL\_Result += f\*f;

}

**return** VMHL\_Result;

}

## 1.6.7 Ссылки

Данная функция приводится в следующих источниках:

1. [9, стр. 4] – GEATbx Examples. Examples of Objective Functions.

Обратите внимание, что иногда под названием Rotated Hyper Ellipsoid встречается (например, [10]) неправильно записанная функция в виде:

$$f(\bar{x}) = \sum_{i=1}^n \sum_{j=1}^j \bar{x}_j^2, \text{ где}$$

Данная функция не является развернутой по своему внешнему виду, поэтому автор склонен считать эту реализацию ошибочной.

## 1.7 Функция Step (модифицированная версия De Jong 3)

### 1.7.1 Описание функции

**Идентификатор:**

MHL\_TestFunction\_StepFunction.

**Наименование:**

Функция Step (модифицированная версия De Jong 3).

**Тип:**

Задача вещественной оптимизации.

**Формула** (целевая функция):

$$f(\bar{x}) = \begin{cases} \sum_{i=1}^n (\text{int}(\bar{x}_i))^2, & \text{если } \sum_{i=1}^n |\text{int}(\bar{x}_i)| \neq 0; \\ \left( \sum_{i=1}^n |\bar{x}_i| \right) - 1, & \text{иначе.} \end{cases}, \quad \text{где} \quad (1.7)$$

$\bar{x} \in X$ ,  $\bar{x}_j \in [Left_j; Right_j]$ ,  $Left_j = -5$ ,  $Right_j = 5$ ,  $j = \overline{1, n}$ .

**Обозначение:**

$\bar{x}$  — вещественный вектор;

$n$  — размерность вещественного вектора.

**Решаемая задача оптимизации:**

$\bar{x}_{min} = \arg \min_{\bar{x} \in X} f(\bar{x})$ .

**Точка минимума:**

$\bar{x}_{min} = (0, 0, \dots, 0)^T$ , то есть  $(\bar{x}_{min})_j = 0$  ( $j = \overline{1, n}$ ).

**Минимум функции:**

$f(\bar{x}_{min}) = -1$ .

**График:**

Рисунок 1.7 на с. 30 стр., 1.8 на с. 30 стр.

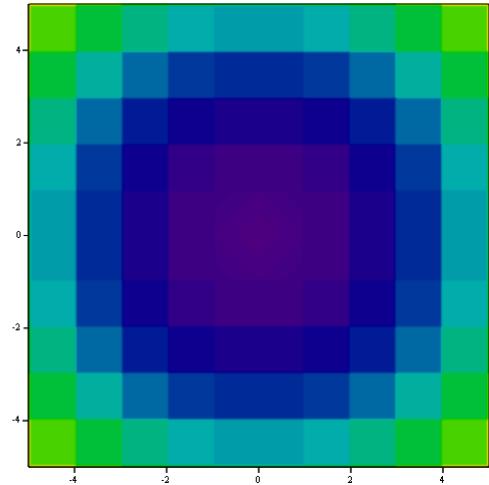
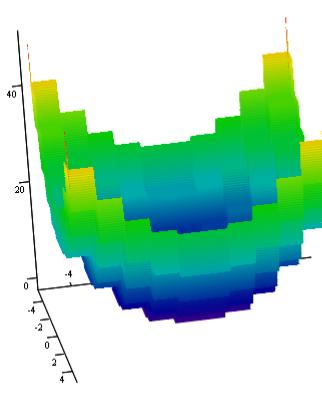


Рисунок 1.7. Функция Step (модифицированная версия De Jong 3)

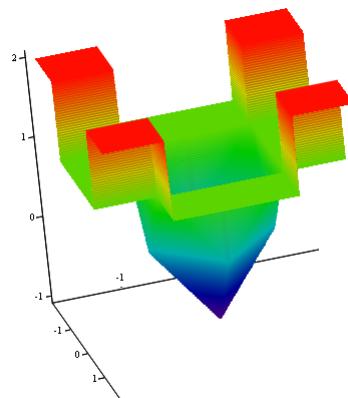


Рисунок 1.8. Функция Step (модифицированная версия De Jong 3) в области около точки минимума

## 1.7.2 Параметры для алгоритмов оптимизации

**Точность вычислений:**  $\varepsilon = 0.025$ .

**Число интервалов, на которые предполагается разбивать каждую компоненту вектора  $\bar{x}$  в пределах своего изменения** (для алгоритмов дискретной оптимизации) :

**Для этого длина бинарной строки для  $x_j$  координаты равна** (для алгоритмов бинарной оптимизации) :

**Замечание:**  $NumberOfParts_j$  выбирается как минимальное число, удовлетворяющее соотношению:

$$NumberOfParts_j = 2^{(k_2)_j} - 1 \geq \frac{10 (Right_j - Left_j)}{\varepsilon}, \text{ где } (k_2)_j \in \mathbb{N}, (j = \overline{1, n}).$$

## 1.7.3 Основная задача и подзадачи

**Изменяемый параметр:**  $n$  — размерность вещественного вектора.

**Значение в основной задаче:**  $n = 2$ .

**Подзадача №2:**  $n = 3$ .

**Подзадача №3:**  $n = 4$ .

**Подзадача №4:**  $n = 5$ .

**Подзадача №5:**  $n = 10$ .

**Подзадача №6:**  $n = 20$ .

**Подзадача №7:**  $n = 30$ .

## 1.7.4 Нахождение ошибки оптимизации

Пусть в результате работы алгоритма оптимизации за  $N$  запусков мы нашли решения  $\bar{x}_{submin}^k$  со значениями целевой функции  $f(\bar{x}_{submin}^k)$  соответственно ( $k = \overline{1, N}$ ). Используем три вида ошибок:

**Надёжность:**

$$R = \frac{\sum_{k=1}^N S(\bar{x}_{submin}^k)}{N}, \text{ где}$$

$$S(\bar{x}_{submin}^k) = \begin{cases} 1, & \text{если } \left| (\bar{x}_{submin}^k)_j - (\bar{x}_{min})_j \right| < \varepsilon, j = \overline{1, n}; \\ 0, & \text{иначе.} \end{cases}$$

**Ошибка по входным параметрам:**

$$E_x = \frac{\sum_{k=1}^N \left( \frac{\sqrt{\sum_{j=1}^n ((\bar{x}_{submin}^k)_j - (\bar{x}_{min})_j)^2}}{n} \right)}{N}.$$

**Ошибка по значениям целевой функции:**

$$E_f = \frac{\sum_{k=1}^N |f(\bar{x}_{submin}^k) - f(\bar{x}_{min})|}{N}.$$

### 1.7.5 Свойства задачи

**Условной или безусловной оптимизации:** Задача безусловной оптимизации.

**Одномерной или многомерной оптимизации:** Многомерной:  $n$ .

**Функция унимодальная или многоэкстремальная:** Функция унимодальная.

**Функция стохастическая или нет:** Функция не стохастическая.

**Особенности:** На большей части множества допустимых решений производная функции равна нулю.

### 1.7.6 Реализация

Реализация функции взята из библиотеки HarrixMathLibrary в разделе «Тестовые функции для оптимизации», которую можно найти по адресу [https://github.com/Harxit/HarrixMathLibrary](https://github.com/Harrix/HarrixMathLibrary).

Код 1.7. Код функции MHL\_TestFunction\_StepFunction

```
double MHL_TestFunction_StepFunction(double *x, int VMHL_N)
{
/*
Функция многих переменных: Функция Step (модифицированная версия De Jong 3).
Тестовая функция вещественной оптимизации.

Входные параметры:
    x - указатель на исходный массив;
    VMHL_N - размер массива x.

Возвращаемое значение:
    Значение тестовой функции в точке x.
*/
    double VMHL_Result=0;

    double H=0;

    for (int i=0;i<VMHL_N;i++)
        H+=fabs(int(x[i]));
}
```

```

if (H!=0)
{
    for (int i=0;i<VMHL_N;i++)
        VMHL_Result+=(int(x[i]))*(int(x[i]));
}
else
{
    for (int i=0;i<VMHL_N;i++)
        VMHL_Result+=fabs(x[i]);
}

return VMHL_Result;
}

```

### 1.7.7 Ссылки

Данная функция приводится в следующих источниках (без дополнительной добавки в области  $\bar{x}_i \in (-1; 1)$ ,  $i = \overline{1, n}$ ):

1. [11, стр. 729] — International Conference on Intelligent Computing: Intelligent computing.

## 1.8 Аддитивная потенциальная функция

### 1.8.1 Описание функции

**Идентификатор:**

MHL\_TestFunction\_AdditivePotential.

**Наименование:**

Аддитивная потенциальная функция.

**Тип:**

Задача вещественной оптимизации.

**Формула** (целевая функция):

$$f(\bar{x}) = z(\bar{x}_1) + z(\bar{x}_2), \text{ где} \quad (1.8)$$

$$z(v) = -\frac{1}{(v-1)^2 + 0.2} - \frac{1}{2(v-2)^2 + 0.15} - \frac{1}{3(v-3)^2 + 0.3},$$

$\bar{x} \in X$ ,  $\bar{x}_j \in [Left_j; Right_j]$ ,  $Left_j = 0$ ,  $Right_j = 4$ ,  $j = \overline{1, n}$ ,  $n = 2$ .

**Обозначение:**

$\bar{x}$  — вещественный вектор;

$n = 2$  — размерность вещественного вектора.

**Решаемая задача оптимизации:**

$\bar{x}_{min} = \arg \min_{\bar{x} \in X} f(\bar{x})$ .

**Точка минимума:**

$\bar{x}_{min} = (2, 2)^T$ , то есть  $(\bar{x}_{min})_j = 2$  ( $j = \overline{1, n}$ ).

**Минимум функции:**

$f(\bar{x}_{min}) = -15.60606060606060606$ .

**График:**

Рисунок 1.9 на с. 34 стр.

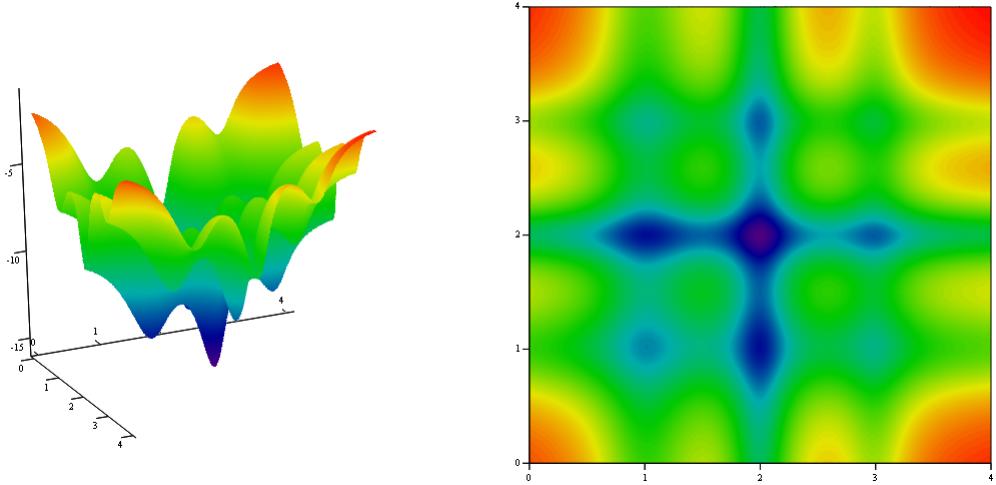


Рисунок 1.9. Аддитивная потенциальная функция

### 1.8.2 Параметры для алгоритмов оптимизации

**Точность вычислений:**

$$\varepsilon = 0.01.$$

**Число интервалов, на которые предполагается разбивать каждую компоненту вектора  $\bar{x}$  в пределах своего изменения** (для алгоритмов дискретной оптимизации) :

**Для этого длина бинарной строки для  $x_j$  координаты равна** (для алгоритмов бинарной оптимизации) :

$$NumberOfParts_j = 4095 \quad (j = \overline{1, n}).$$

**Замечание:**  $NumberOfParts_j$  выбирается как минимальное число, удовлетворяющее соотношению:

$$NumberOfParts_j = 2^{(k_2)_j} - 1 \geq \frac{10 (Right_j - Left_j)}{\varepsilon}, \text{ где } (k_2)_j \in \mathbb{N}, (j = \overline{1, n}).$$

### 1.8.3 Основная задача и подзадачи

**Изменяемый параметр:**

$n$  — размерность вещественного вектора.

**Значение в основной задаче:**

$$n = 2.$$

### 1.8.4 Нахождение ошибки оптимизации

Пусть в результате работы алгоритма оптимизации за  $N$  запусков мы нашли решения  $\bar{x}_{submin}^k$  со значениями целевой функции  $f(\bar{x}_{submin}^k)$  соответственно ( $k = \overline{1, N}$ ). Используем три вида ошибок:

**Надёжность:**

$$R = \frac{\sum_{k=1}^N S(\bar{x}_{submin}^k)}{N}, \text{ где}$$
$$S(\bar{x}_{submin}^k) = \begin{cases} 1, & \text{если } \left| (\bar{x}_{submin}^k)_j - (\bar{x}_{min})_j \right| < \varepsilon, j = \overline{1, n}; \\ 0, & \text{иначе.} \end{cases}$$

**Ошибка по входным параметрам:**

$$E_x = \frac{\sum_{k=1}^N \left( \frac{\sqrt{\sum_{j=1}^n ((\bar{x}_{submin}^k)_j - (\bar{x}_{min})_j)^2}}{n} \right)}{N}.$$

**Ошибка по значениям целевой функции:**

$$E_f = \frac{\sum_{k=1}^N |f(\bar{x}_{submin}^k) - f(\bar{x}_{min})|}{N}.$$

## 1.8.5 Свойства задачи

**Условной или безусловной оптимизации:** Задача безусловной оптимизации.

**Одномерной или многомерной оптимизации:** Многомерной (двумерной).

**Функция унимодальная или многоэкстремальная:** Функция многоэкстремальная.

**Функция стохастическая или нет:** Функция не стохастическая.

**Особенности:** Нет.

## 1.8.6 Реализация

Реализация функции взята из библиотеки HarrixMathLibrary в разделе «Тестовые функции для оптимизации», которую можно найти по адресу <https://github.com/Harrix/HarrixMathLibrary>.

Код 1.8. Код функции MHL\_TestFunction\_AdditivePotential

```
double MHL_TestFunction_AdditivePotential(double x, double y)
{
    /*

Функция двух переменных: аддитивная потенциальная функция.

Тестовая функция вещественной оптимизации.

Входные параметры:
    x - первая вещественная переменная;
    y - вторая вещественная переменная.

Возвращаемое значение:
    Значение тестовой функции в точке (x, y).
}
```

```

*/
double VMHL_Result;
double z1=-(1./((x-1.)*(x-1.)+0.2))-(1./(2.*(x-2.)*(x-2.)+0.15))-(1./(3.*(x-3.)*(x
-3.))+0.3));
double z2=-(1./((y-1.)*(y-1.)+0.2))-(1./(2.*(y-2.)*(y-2.)+0.15))-(1./(3.*(y-3.)*(y
-3.))+0.3));
VMHL_Result=z1+z2;
return VMHL_Result;
}

```

## 1.8.7 Ссылки

Данная функция приводится в следующих источниках:

1. [12, стр. 33] — Эволюционные методы моделирования и оптимизации сложных систем.

## 1.9 Функция Химмельблау

### 1.9.1 Описание функции

**Идентификатор:** MHL\_TestFunction\_Himmelblau.

**Наименование:** Функция Химмельблау.

**Тип:** Задача вещественной оптимизации.

**Формула** (целевая функция):

$$f(\bar{x}) = (\bar{x}_1^2 + \bar{x}_2 - 11)^2 + (\bar{x}_1 + \bar{x}_2^2 - 7)^2, \text{ где} \quad (1.9)$$

$\bar{x} \in X, \bar{x}_j \in [Left_j; Right_j], Left_j = -5, Right_j = 5, j = \overline{1, n}, n = 2.$

**Обозначение:**  $\bar{x}$  — вещественный вектор;

$n = 2$  — размерность вещественного вектора.

**Решаемая задача оптимизации:**  $\bar{x}_{min} = \arg \min_{\bar{x} \in X} f(\bar{x}).$

**Точки минимума:**  $\bar{x}_{min}^1 = (3, 2)^T,$

$$\bar{x}_{min}^2 \approx (-2.8051183, 3.131312)^T$$

$$\bar{x}_{min}^3 \approx (-3.779310, -3.283186)^T$$

$$\bar{x}_{min}^4 \approx (3.584428, -1.848126)^T.$$

**Минимум функции:**  $f(\bar{x}_{min}^i) = 0, i = \overline{1, 4}.$

**График:** Рисунок 1.10 на с. 37 стр.

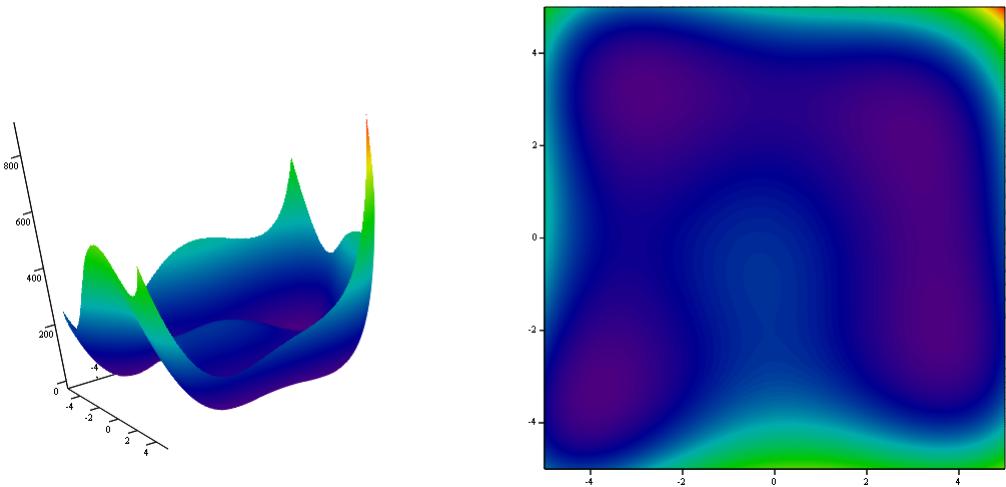


Рисунок 1.10. Функция Химмельблау

### 1.9.2 Параметры для алгоритмов оптимизации

**Точность вычислений:**  $\varepsilon = 0.025.$

**Число интервалов, на которые предполагается разбивать каждую компоненту вектора  $\bar{x}$  в пределах своего изменения** (для алгоритмов дискретной оптимизации) :

**Для этого длина бинарной строки для  $x_j$  координаты равна** (для алгоритмов бинарной оптимизации) :  $(k_2)_j = 12 (j = \overline{1, n}).$

**Замечание:**  $NumberOfParts_j$  выбирается как минимальное число, удовлетворяющее соотношению:

$$NumberOfParts_j = 2^{(k_2)_j} - 1 \geq \frac{10 (Right_j - Left_j)}{\varepsilon}, \text{ где } (k_2)_j \in \mathbb{N}, (j = \overline{1, n}).$$

### 1.9.3 Основная задача и подзадачи

**Изменяемый параметр:**  $n$  — размерность вещественного вектора.

**Значение в основной задаче:**  $n = 2.$

### 1.9.4 Нахождение ошибки оптимизации

**Внимание!** В отличии от других функций формулы нахождения ошибок другие, так как есть несколько идентичных по значению целевой функции глобальных минимумов.

Пусть в результате работы алгоритма оптимизации за  $N$  запусков мы нашли решения  $\bar{x}_{submin}^k$  со значениями целевой функции  $f(\bar{x}_{submin}^k)$  соответственно ( $k = \overline{1, N}$ ). Используем три вида ошибок:

**Надёжность:**

$$R = \frac{\sum_{k=1}^N S(\bar{x}_{submin}^k)}{N}, \text{ где}$$

$$S(\bar{x}_{submin}^k) = \begin{cases} 1, & \text{если } \left| (\bar{x}_{submin}^k)_j - (\bar{x}_{min}^1)_j \right| < \varepsilon, j = \overline{1, n}; \\ 1, & \text{если } \left| (\bar{x}_{submin}^k)_j - (\bar{x}_{min}^2)_j \right| < \varepsilon, j = \overline{1, n}; \\ 1, & \text{если } \left| (\bar{x}_{submin}^k)_j - (\bar{x}_{min}^3)_j \right| < \varepsilon, j = \overline{1, n}; \\ 1, & \text{если } \left| (\bar{x}_{submin}^k)_j - (\bar{x}_{min}^4)_j \right| < \varepsilon, j = \overline{1, n}; \\ 0, & \text{иначе.} \end{cases}$$

**Ошибка по входным параметрам:**

$$E_x = \min_{i=1,4} \left\{ \frac{\sum_{k=1}^N \left( \frac{\sqrt{\sum_{j=1}^n ((\bar{x}_{submin}^k)_j - (\bar{x}_{min}^i)_j)^2}}{n} \right)}{N} \right\}.$$

**Ошибка по значениям целевой функции:** (без изменений)

$$E_f = \frac{\sum_{k=1}^N |f(\bar{x}_{submin}^k) - f(\bar{x}_{min})|}{N}.$$

### 1.9.5 Свойства задачи

**Условной или безусловной оптимизации:** Задача безусловной оптимизации.

**Одномерной или многомерной оптимизации:** Многомерной: (двумерной).

**Функция унимодальная или многоэкстремальная:** Функция многоэкстремальная.

**Функция стохастическая или нет:** Функция не стохастическая.

**Особенности:** Есть 4 глобальных минимума.

### 1.9.6 Реализация

Реализация функции взята из библиотеки HarrixMathLibrary в разделе «Тестовые функции для оптимизации», которую можно найти по адресу <https://github.com/Harrix/HarrixMathLibrary>.

Код 1.9. Код функции MHL\_TestFunction\_Himmelblau

```
double MHL_TestFunction_Himmelblau(double x, double y)
```

```

{
/*
Функция двух переменных: функция Химмельблау.
Тестовая функция вещественной оптимизации.
Входные параметры:
    x - первая вещественная переменная;
    y - вторая вещественная переменная.
Возвращаемое значение:
    Значение тестовой функции в точке (x,y).
*/
double VMHL_Result;
VMHL_Result=(x*x+y-11)*(x*x+y-11)+(x+y*y-7)*(x+y*y-7);
return VMHL_Result;
}

```

## 1.9.7 Ссылки

Данная функция приводится в следующих источниках:

1. [13] – Himmelblau's function.
2. [14] – Minimization of the Himmelblau Function.

## 1.10 Функция Катникова

### 1.10.1 Описание функции

**Идентификатор:**

MHL\_TestFunction\_Katnikov.

**Наименование:**

Функция Катникова.

**Тип:**

Задача вещественной оптимизации.

**Формула** (целевая функция):

$$f(\bar{x}) = 0.5 (\bar{x}_1^2 + \bar{x}_2^2) \left( 2A + A \cos(1.5\bar{x}_1) \cos(3.14\bar{x}_2) + A \cos(\sqrt{5}\bar{x}_1) \cos(3.5\bar{x}_2) \right), \text{ где } \quad (1.10)$$

$\bar{x} \in X$ ,  $\bar{x}_j \in [Left_j; Right_j]$ ,  $Left_j = -5$ ,  $Right_j = 5$ ,  $j = \overline{1, n}$ ,  $n = 2$ ,  $A = 0.8$ .

**Обозначение:**

$\bar{x}$  – вещественный вектор;

$n = 2$  – размерность вещественного вектора.

**Решаемая задача оптимизации:**

$\bar{x}_{min} = \arg \min_{\bar{x} \in X} f(\bar{x})$ .

**Точка минимума:**

$\bar{x}_{min} = (0, 0)^T$ , то есть  $(\bar{x}_{min})_j = 0$  ( $j = \overline{1, n}$ ).

**Минимум функции:**

$f(\bar{x}_{min}) = 0$ .

**График:**

Рисунок 1.11 на с. 40 стр.

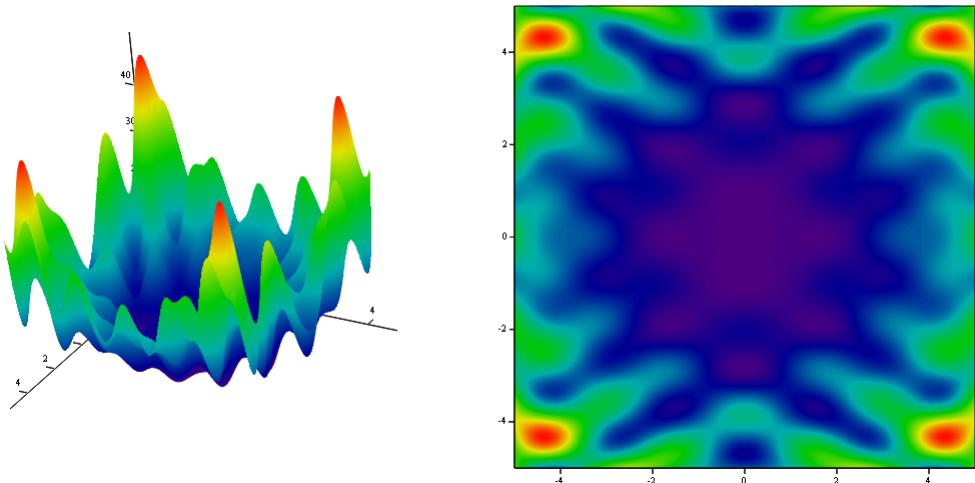


Рисунок 1.11. Функция Катникова

### 1.10.2 Параметры для алгоритмов оптимизации

**Точность вычислений:**  $\varepsilon = 0.025.$

**Число интервалов, на которые предполагается разбивать каждую компоненту вектора  $\bar{x}$  в пределах своего изменения** (для алгоритмов дискретной оптимизации) :

**Для этого длина бинарной строки для  $x_j$  координаты равна** (для алгоритмов бинарной оптимизации) :  $(k_2)_j = 12 (j = \overline{1, n}).$

**Замечание:**  $NumberOfParts_j$  выбирается как минимальное число, удовлетворяющее соотношению:

$$NumberOfParts_j = 2^{(k_2)_j} - 1 \geq \frac{10 (Right_j - Left_j)}{\varepsilon}, \text{ где } (k_2)_j \in \mathbb{N}, (j = \overline{1, n}).$$

### 1.10.3 Основная задача и подзадачи

**Изменяемый параметр:**  $n$  — размерность вещественного вектора.

**Значение в основной задаче:**  $n = 2.$

### 1.10.4 Нахождение ошибки оптимизации

Пусть в результате работы алгоритма оптимизации за  $N$  запусков мы нашли решения  $\bar{x}_{submin}^k$  со значениями целевой функции  $f(\bar{x}_{submin}^k)$  соответственно ( $k = \overline{1, N}$ ). Используем три вида ошибок:

**Надёжность:**

$$R = \frac{\sum_{k=1}^N S(\bar{x}_{submin}^k)}{N}, \text{ где}$$
$$S(\bar{x}_{submin}^k) = \begin{cases} 1, & \text{если } \left| (\bar{x}_{submin}^k)_j - (\bar{x}_{min})_j \right| < \varepsilon, j = \overline{1, n}; \\ 0, & \text{иначе.} \end{cases}$$

**Ошибка по входным параметрам:**

$$E_x = \frac{\sum_{k=1}^N \left( \frac{\sqrt{\sum_{j=1}^n ((\bar{x}_{submin}^k)_j - (\bar{x}_{min})_j)^2}}{n} \right)}{N}.$$

**Ошибка по значениям целевой функции:**

$$E_f = \frac{\sum_{k=1}^N |f(\bar{x}_{submin}^k) - f(\bar{x}_{min})|}{N}.$$

## 1.10.5 Свойства задачи

**Условной или безусловной оптимизации:** Задача безусловной оптимизации.

**Одномерной или многомерной оптимизации:** Многомерной: (двумерной).

**Функция унимодальная или многоэкстремальная:** Функция многоэкстремальная.

**Функция стохастическая или нет:** Функция не стохастическая.

**Особенности:** Нет.

## 1.10.6 Реализация

Реализация функции взята из библиотеки HarrixMathLibrary в разделе «Тестовые функции для оптимизации», которую можно найти по адресу <https://github.com/Harrix/HarrixMathLibrary>.

Код 1.10. Код функции MHL\_TestFunction\_Katnikov

```
double MHL_TestFunction_Katnikov(double x, double y)
{
    /*

Функция двух переменных: функция Катникова.

Тестовая функция вещественной оптимизации.

Входные параметры:
    x - первая вещественная переменная;
    y - вторая вещественная переменная.

Возвращаемое значение:
    Значение тестовой функции в точке (x, y).
}
```

```

*/
double VMHL_Result;
double A=0.8;
VMHL_Result=0.5*(x*x+y*y)*(2*A+A*cos(1.5*x)*cos(3.14*y)+A*cos(sqrt(5)*x)*cos(3.5*y));
return VMHL_Result;
}

```

## 1.10.7 Ссылки

Данная функция приводится в следующих источниках:

1. [12, стр. 31] — Эволюционные методы моделирования и оптимизации сложных систем.

## 1.11 Функция Multiextremal3

### 1.11.1 Описание функции

**Идентификатор:**

MHL\_TestFunction\_Multiextremal3.

**Наименование:**

Функция Multiextremal3.

**Тип:**

Задача вещественной оптимизации.

**Формула** (целевая функция):

$$f(\bar{x}) = \bar{x}_1^2 |\sin(2\bar{x}_1)| + \bar{x}_2^2 |\sin(2\bar{x}_2)| - \frac{1}{5\bar{x}_1^2 + 5\bar{x}_2^2 + 0.2} + 5, \text{ где } \quad (1.11)$$

$\bar{x} \in X$ ,  $\bar{x}_j \in [Left_j; Right_j]$ ,  $Left_j = -5$ ,  $Right_j = 5$ ,  $j = \overline{1, n}$ ,  $n = 2$ .

**Обозначение:**

$\bar{x}$  — вещественный вектор;

$n = 2$  — размерность вещественного вектора.

**Решаемая задача оптимизации:**

$\bar{x}_{min} = \arg \min_{\bar{x} \in X} f(\bar{x})$ .

**Точка минимума:**

$\bar{x}_{min} = (0, 0)^T$ , то есть  $(\bar{x}_{min})_j = 0$  ( $j = \overline{1, n}$ ).

**Минимум функции:**

$f(\bar{x}_{min}) = 0$ .

**График:**

Рисунок 1.12 на с. 43 стр.

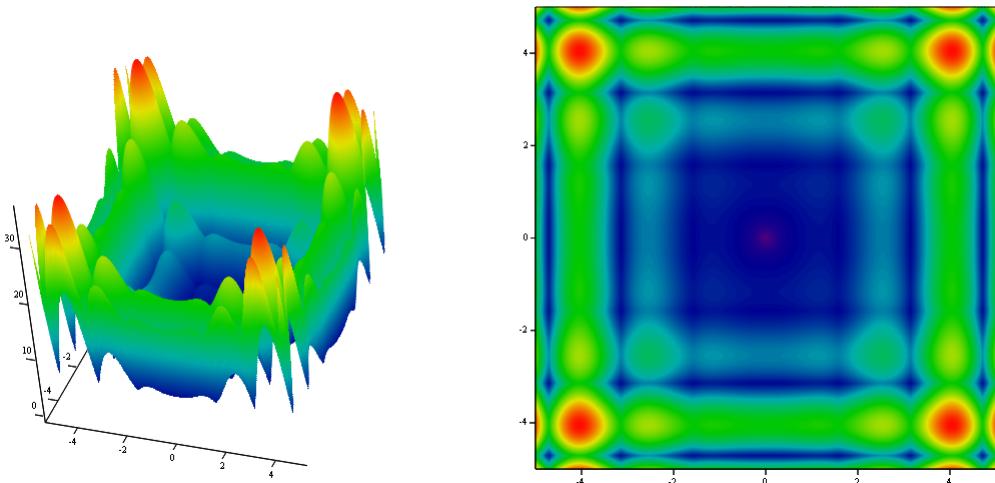


Рисунок 1.12. Функция Multiextremal3

### 1.11.2 Параметры для алгоритмов оптимизации

**Точность вычислений:**

$$\varepsilon = 0.025.$$

**Число интервалов, на которые предполагается разбивать каждую компоненту вектора  $\bar{x}$  в пределах своего изменения** (для алгоритмов дискретной оптимизации) :

**Для этого длина бинарной строки для  $x_j$  координаты равна** (для алгоритмов бинарной оптимизации) :

$$Number\text{Of} \text{ } Parts_j = 4095 \ (j = \overline{1, n}).$$

**Замечание:**  $Number\text{Of} \text{ } Parts_j$  выбирается как минимальное число, удовлетворяющее соотношению:

$$Number\text{Of} \text{ } Parts_j = 2^{(k_2)_j} - 1 \geq \frac{10 (Right_j - Left_j)}{\varepsilon}, \text{ где } (k_2)_j \in \mathbb{N}, (j = \overline{1, n}).$$

### 1.11.3 Основная задача и подзадачи

**Изменяемый параметр:**

$n$  — размерность вещественного вектора.

**Значение в основной задаче:**

$$n = 2.$$

### 1.11.4 Нахождение ошибки оптимизации

Пусть в результате работы алгоритма оптимизации за  $N$  запусков мы нашли решения  $\bar{x}_{submin}^k$  со значениями целевой функции  $f(\bar{x}_{submin}^k)$  соответственно ( $k = \overline{1, N}$ ). Используем три вида ошибок:

**Надёжность:**

$$R = \frac{\sum_{k=1}^N S(\bar{x}_{submin}^k)}{N}, \text{ где}$$
$$S(\bar{x}_{submin}^k) = \begin{cases} 1, & \text{если } \left| (\bar{x}_{submin}^k)_j - (\bar{x}_{min})_j \right| < \varepsilon, j = \overline{1, n}; \\ 0, & \text{иначе.} \end{cases}$$

**Ошибка по входным параметрам:**

$$E_x = \frac{\sum_{k=1}^N \left( \frac{\sqrt{\sum_{j=1}^n ((\bar{x}_{submin}^k)_j - (\bar{x}_{min})_j)^2}}{n} \right)}{N}.$$

**Ошибка по значениям целевой функции:**

$$E_f = \frac{\sum_{k=1}^N |f(\bar{x}_{submin}^k) - f(\bar{x}_{min})|}{N}.$$

## 1.11.5 Свойства задачи

**Условной или безусловной оптимизации:** Задача безусловной оптимизации.

**Одномерной или многомерной оптимизации:** Многомерной: (двумерной).

**Функция унимодальная или многоэкстремальная:** Функция многоэкстремальная.

**Функция стохастическая или нет:** Функция не стохастическая.

**Особенности:** Нет.

## 1.11.6 Реализация

Реализация функции взята из библиотеки HarrixMathLibrary в разделе «Тестовые функции для оптимизации», которую можно найти по адресу <https://github.com/Harrix/HarrixMathLibrary>.

Код 1.11. Код функции MHL\_TestFunction\_Multiextremal3

```
double MHL_TestFunction_Multiextremal3(double x, double y)
{
    /*
Функция двух переменных: функция Multiextremal3.
Тестовая функция вещественной оптимизации.
Входные параметры:
    x - первая вещественная переменная;
    y - вторая вещественная переменная.
Возвращаемое значение:
    Значение тестовой функции в точке (x, y).
}
```

```

*/
double VMHL_Result;
VMHL_Result=x*x*fabs(sin(2.*x))+y*y*fabs(sin(2.*y))-1./(5.*x*x+5.*y*y+0.2)+5.;
return VMHL_Result;
}

```

## 1.11.7 Ссылки

Данная функция приводится в следующих источниках:

1. [12, стр. 31] — Эволюционные методы моделирования и оптимизации сложных систем.

## 1.12 Функция Multiextremal4

### 1.12.1 Описание функции

**Идентификатор:**

MHL\_TestFunction\_Multiextremal4.

**Наименование:**

Функция Multiextremal4.

**Тип:**

Задача вещественной оптимизации.

**Формула** (целевая функция):

$$f(\bar{x}) = 0.5 (\bar{x}_1^2 + \bar{x}_1 \bar{x}_2 + \bar{x}_2^2) (1 + 0.5 \cos(1.5\bar{x}_1) \cos(3.2\bar{x}_1\bar{x}_2) \cos(3.14\bar{x}_2) + 0.5 \cos(2.2\bar{x}_1) \cos(4.8\bar{x}_1\bar{x}_2) \cos(3.5\bar{x}_2)), \text{ где} \quad (1.12)$$

$\bar{x} \in X$ ,  $\bar{x}_j \in [Left_j; Right_j]$ ,  $Left_j = 0$ ,  $Right_j = 4$ ,  $j = \overline{1, n}$ ,  $n = 2$ .

**Обозначение:**

$\bar{x}$  — вещественный вектор;

$n = 2$  — размерность вещественного вектора.

**Решаемая задача оптимизации:**

$\bar{x}_{min} = \arg \min_{\bar{x} \in X} f(\bar{x})$ .

**Точка минимума:**

$\bar{x}_{min} = (0, 0)^T$ , то есть  $(\bar{x}_{min})_j = 0$  ( $j = \overline{1, n}$ ).

**Минимум функции:**

$f(\bar{x}_{min}) = 0$ .

**График:**

Рисунок 1.13 на с. 46 стр.

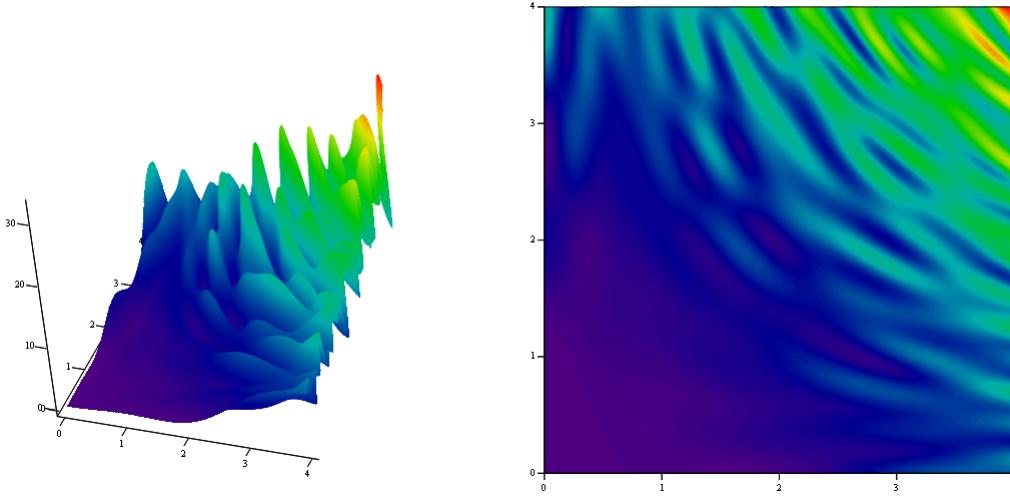


Рисунок 1.13. Функция Multiextremal4

## 1.12.2 Параметры для алгоритмов оптимизации

**Точность вычислений:**

$$\varepsilon = 0.01.$$

**Число интервалов, на которые предполагается разбивать каждую компоненту вектора  $\bar{x}$  в пределах своего изменения** (для алгоритмов дискретной оптимизации) :

**Для этого длина бинарной строки для  $x_j$  координаты равна** (для алгоритмов бинарной оптимизации) :

$$Number\text{Of}Parts_j = 4095 \quad (j = \overline{1, n}).$$

**Замечание:**  $Number\text{Of}Parts_j$  выбирается как минимальное число, удовлетворяющее соотношению:

$$Number\text{Of}Parts_j = 2^{(k_2)_j} - 1 \geq \frac{10 (Right_j - Left_j)}{\varepsilon}, \text{ где } (k_2)_j \in \mathbb{N}, (j = \overline{1, n}).$$

## 1.12.3 Основная задача и подзадачи

**Изменяемый параметр:**

$n$  — размерность вещественного вектора.

**Значение в основной задаче:**

$$n = 2.$$

## 1.12.4 Нахождение ошибки оптимизации

Пусть в результате работы алгоритма оптимизации за  $N$  запусков мы нашли решения  $\bar{x}_{submin}^k$  со значениями целевой функции  $f(\bar{x}_{submin}^k)$  соответственно ( $k = \overline{1, N}$ ). Используем три вида ошибок:

**Надёжность:**

$$R = \frac{\sum_{k=1}^N S(\bar{x}_{submin}^k)}{N}, \text{ где}$$
$$S(\bar{x}_{submin}^k) = \begin{cases} 1, & \text{если } \left| (\bar{x}_{submin}^k)_j - (\bar{x}_{min})_j \right| < \varepsilon, j = \overline{1, n}; \\ 0, & \text{иначе.} \end{cases}$$

**Ошибка по входным параметрам:**

$$E_x = \frac{\sum_{k=1}^N \left( \frac{\sqrt{\sum_{j=1}^n ((\bar{x}_{submin}^k)_j - (\bar{x}_{min})_j)^2}}{n} \right)}{N}.$$

**Ошибка по значениям целевой функции:**

$$E_f = \frac{\sum_{k=1}^N |f(\bar{x}_{submin}^k) - f(\bar{x}_{min})|}{N}.$$

## 1.12.5 Свойства задачи

**Условной или безусловной оптимизации:** Задача безусловной оптимизации.

**Одномерной или многомерной оптимизации:** Многомерной: (двумерной).

**Функция унимодальная или многоэкстремальная:** Функция многоэкстремальная.

**Функция стохастическая или нет:** Функция не стохастическая.

**Особенности:** Нет.

## 1.12.6 Реализация

Реализация функции взята из библиотеки HarrixMathLibrary в разделе «Тестовые функции для оптимизации», которую можно найти по адресу <https://github.com/Harrix/HarrixMathLibrary>.

Код 1.12. Код функции MHL\_TestFunction\_Multiextremal4

```
double MHL_TestFunction_Multiextremal4(double x, double y)
{
    /*
Функция двух переменных: функция Multiextremal4.
Тестовая функция вещественной оптимизации.
Входные параметры:
    x - первая вещественная переменная;
    y - вторая вещественная переменная.
Возвращаемое значение:
    Значение тестовой функции в точке (x, y).
}
```

```

*/
double VMHL_Result;
VMHL_Result=0.5*(x*x+x*y+y*y)*(1.+0.5*cos(1.5*x)*cos(3.2*x*y)*cos(3.14*y)+0.5*cos(2.2*x)*cos(4.8*x*y)*cos(3.5*y));
return VMHL_Result;
}

```

## 1.12.7 Ссылки

Данная функция приводится в следующих источниках:

1. [12, стр. 31] — Эволюционные методы моделирования и оптимизации сложных систем.

## 1.13 Мультипликативная потенциальная функция

### 1.13.1 Описание функции

<b>Идентификатор:</b>	MHL_TestFunction_MultiplicativePotential.
<b>Наименование:</b>	Мультипликативная потенциальная функция.
<b>Тип:</b>	Задача вещественной оптимизации.
<b>Формула</b> (целевая функция):	

$$f(\bar{x}) = -z(\bar{x}_1) \cdot z(\bar{x}_2), \text{ где} \quad (1.13)$$

$$z(v) = -\frac{1}{(v-1)^2 + 0.2} - \frac{1}{2(v-2)^2 + 0.15} - \frac{1}{3(v-3)^2 + 0.3},$$

$$\bar{x} \in X, \bar{x}_j \in [Left_j; Right_j], Left_j = 0, Right_j = 4, j = \overline{1, n}, n = 2.$$

<b>Обозначение:</b>	$\bar{x}$ — вещественный вектор;
	$n = 2$ — размерность вещественного вектора.
<b>Решаемая задача оптимизации:</b>	$\bar{x}_{min} = \arg \min_{\bar{x} \in X} f(\bar{x}).$
<b>Точка минимума:</b>	$\bar{x}_{min} = (2, 2)^T$ , то есть $(\bar{x}_{min})_j = 2$ ( $j = \overline{1, n}$ ).
<b>Минимум функции:</b>	$f(\bar{x}_{min}) = -60.8872819100091.$
<b>График:</b>	Рисунок 1.14 на с. 49 стр.

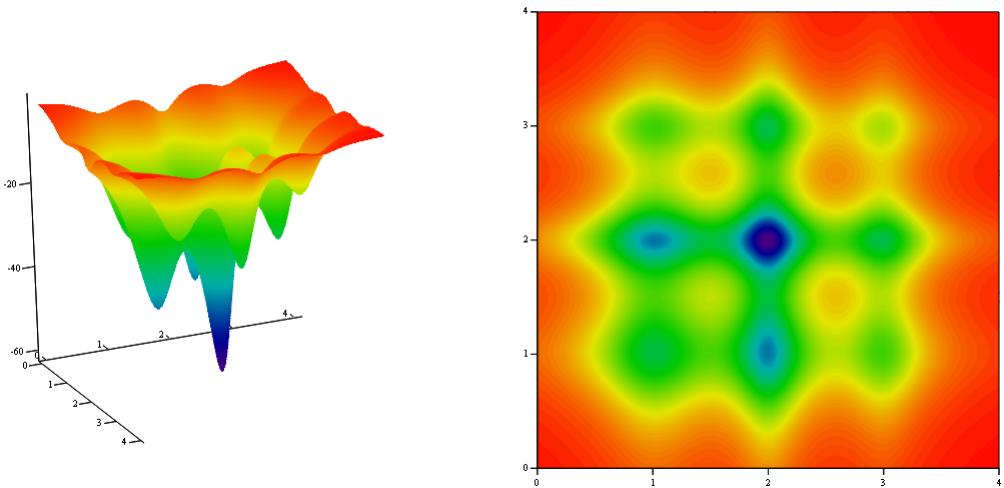


Рисунок 1.14. Мультипликативная потенциальная функция

### 1.13.2 Параметры для алгоритмов оптимизации

**Точность вычислений:**

$$\varepsilon = 0.01.$$

**Число интервалов, на которые предполагается разбивать каждую компоненту вектора  $\bar{x}$  в пределах своего изменения** (для алгоритмов дискретной оптимизации) :

**Для этого длина бинарной строки для  $x_j$  координаты равна** (для алгоритмов бинарной оптимизации) :

$$NumberOfParts_j = 4095 \quad (j = \overline{1, n}).$$

**Замечание:**  $NumberOfParts_j$  выбирается как минимальное число, удовлетворяющее соотношению:

$$NumberOfParts_j = 2^{(k_2)_j} - 1 \geq \frac{10 (Right_j - Left_j)}{\varepsilon}, \text{ где } (k_2)_j \in \mathbb{N}, (j = \overline{1, n}).$$

### 1.13.3 Основная задача и подзадачи

**Изменяемый параметр:**

$n$  — размерность вещественного вектора.

**Значение в основной задаче:**

$$n = 2.$$

### 1.13.4 Нахождение ошибки оптимизации

Пусть в результате работы алгоритма оптимизации за  $N$  запусков мы нашли решения  $\bar{x}_{submin}^k$  со значениями целевой функции  $f(\bar{x}_{submin}^k)$  соответственно ( $k = \overline{1, N}$ ). Используем три вида ошибок:

**Надёжность:**

$$R = \frac{\sum_{k=1}^N S(\bar{x}_{submin}^k)}{N}, \text{ где}$$
$$S(\bar{x}_{submin}^k) = \begin{cases} 1, & \text{если } \left| (\bar{x}_{submin}^k)_j - (\bar{x}_{min})_j \right| < \varepsilon, j = \overline{1, n}; \\ 0, & \text{иначе.} \end{cases}$$

**Ошибка по входным параметрам:**

$$E_x = \frac{\sum_{k=1}^N \left( \frac{\sqrt{\sum_{j=1}^n ((\bar{x}_{submin}^k)_j - (\bar{x}_{min})_j)^2}}{n} \right)}{N}.$$

**Ошибка по значениям целевой функции:**

$$E_f = \frac{\sum_{k=1}^N |f(\bar{x}_{submin}^k) - f(\bar{x}_{min})|}{N}.$$

### 1.13.5 Свойства задачи

**Условной или безусловной оптимизации:** Задача безусловной оптимизации.

**Одномерной или многомерной оптимизации:** Многомерной: (двумерной).

**Функция унимодальная или многоэкстремальная:** Функция многоэкстремальная.

**Функция стохастическая или нет:** Функция не стохастическая.

**Особенности:** Нет.

### 1.13.6 Реализация

Реализация функции взята из библиотеки HarrixMathLibrary в разделе «Тестовые функции для оптимизации», которую можно найти по адресу <https://github.com/Harrix/HarrixMathLibrary>.

Код 1.13. Код функции MHL\_TestFunction\_MultiplicativePotential

```
double MHL_TestFunction_MultiplicativePotential(double x, double y)
{
    /*
```

*Функция двух переменных: мультипликативная потенциальная функция.*

*Тестовая функция вещественной оптимизации.*

*Входные параметры:*

*x – первая вещественная переменная;*

*y – вторая вещественная переменная.*

*Возвращаемое значение:*

*Значение тестовой функции в точке (x, y).*

```

*/
double VMHL_Result;
double z1=-(1./((x-1.)*(x-1.)+0.2))-(1./(2.*(x-2.)*(x-2.)+0.15))-(1./(3.*(x-3.)*(x
-3.))+0.3));
double z2=-(1./((y-1.)*(y-1.)+0.2))-(1./(2.*(y-2.)*(y-2.)+0.15))-(1./(3.*(y-3.)*(y
-3.))+0.3));
VMHL_Result=-z1*z2;
return VMHL_Result;
}

```

## 1.13.7 Ссылки

Данная функция приводится в следующих источниках:

1. [12, стр. 32] — Эволюционные методы моделирования и оптимизации сложных систем.

## 1.14 Функция Rana

### 1.14.1 Описание функции

**Идентификатор:** MHL\_TestFunction\_Rana.

**Наименование:** Функция Rana.

**Тип:** Задача вещественной оптимизации.

**Формула** (целевая функция):

$$f(\bar{x}) = \bar{x}_1 \sin \left( \sqrt{|\bar{x}_2 + 1 - \bar{x}_1|} \right) \cos \left( \sqrt{|\bar{x}_2 + 1 + \bar{x}_1|} \right) + \quad (1.14)$$

$$+ (\bar{x}_2 + 1) \cos \left( \sqrt{|\bar{x}_2 + 1 - \bar{x}_1|} \right) \sin \left( \sqrt{|\bar{x}_2 + 1 + \bar{x}_1|} \right), \text{ где} \quad (1.15)$$

$\bar{x} \in X$ ,  $\bar{x}_j \in [Left_j; Right_j]$ ,  $Left_j = -512$ ,  $Right_j = 512$ ,  $j = \overline{1, n}$ ,  $n = 2$ .

**Обозначение:**  $\bar{x}$  — вещественный вектор;

$n = 2$  — размерность вещественного вектора.

**Решаемая задача оптимизации:**  $\bar{x}_{min} = \arg \min_{\bar{x} \in X} f(\bar{x})$ .

**Точка минимума:**  $\bar{x}_{min} = (-488.6326, 512)^T$ .

**Минимум функции:**  $f(\bar{x}_{min}) = -511.7328819$ .

**График:** Рисунок 1.15 на с. 52 стр.

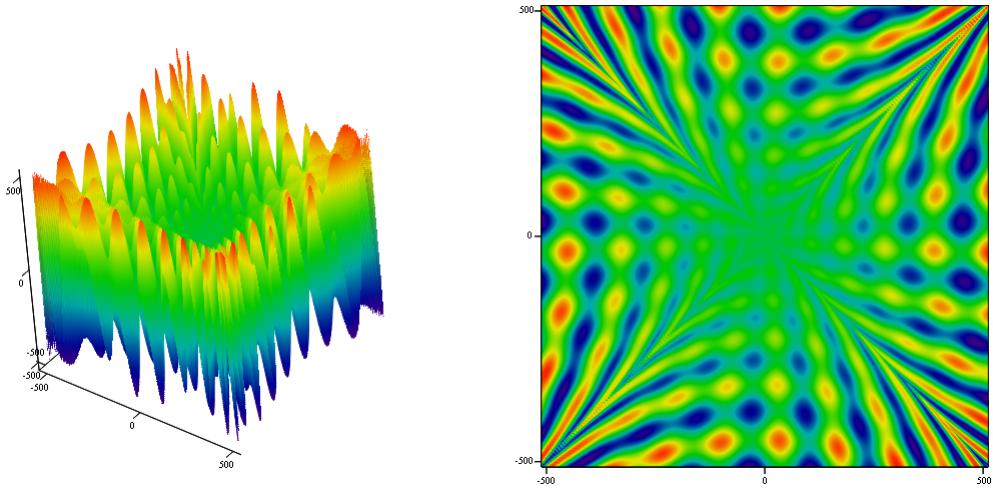


Рисунок 1.15. Функция Rana

## 1.14.2 Параметры для алгоритмов оптимизации

**Точность вычислений:**

$$\varepsilon = 2.5.$$

**Число интервалов, на которые предполагается разбивать каждую компоненту вектора  $\bar{x}$  в пределах своего изменения** (для алгоритмов дискретной оптимизации) :

$$NumberOfParts_j = 4095 \quad (j = \overline{1, n}).$$

**Для этого длина бинарной строки для  $x_j$  координаты равна** (для алгоритмов бинарной оптимизации) :

$$(k_2)_j = 12 \quad (j = \overline{1, n}).$$

**Замечание:**  $NumberOfParts_j$  выбирается как минимальное число, удовлетворяющее соотношению:

$$NumberOfParts_j = 2^{(k_2)_j} - 1 \geq \frac{10 (Right_j - Left_j)}{\varepsilon}, \text{ где } (k_2)_j \in \mathbb{N}, (j = \overline{1, n}).$$

## 1.14.3 Основная задача и подзадачи

**Изменяемый параметр:**

$n$  — размерность вещественного вектора.

**Значение в основной задаче:**

$$n = 2.$$

## 1.14.4 Нахождение ошибки оптимизации

Пусть в результате работы алгоритма оптимизации за  $N$  запусков мы нашли решения  $\bar{x}_{submin}^k$  со значениями целевой функции  $f(\bar{x}_{submin}^k)$  соответственно ( $k = \overline{1, N}$ ). Используем три вида ошибок:

**Надёжность:**

$$R = \frac{\sum_{k=1}^N S(\bar{x}_{submin}^k)}{N}, \text{ где}$$
$$S(\bar{x}_{submin}^k) = \begin{cases} 1, & \text{если } \left| (\bar{x}_{submin}^k)_j - (\bar{x}_{min})_j \right| < \varepsilon, j = \overline{1, n}; \\ 0, & \text{иначе.} \end{cases}$$

**Ошибка по входным параметрам:**

$$E_x = \frac{\sum_{k=1}^N \left( \frac{\sqrt{\sum_{j=1}^n ((\bar{x}_{submin}^k)_j - (\bar{x}_{min})_j)^2}}{n} \right)}{N}.$$

**Ошибка по значениям целевой функции:**

$$E_f = \frac{\sum_{k=1}^N |f(\bar{x}_{submin}^k) - f(\bar{x}_{min})|}{N}.$$

## 1.14.5 Свойства задачи

**Условной или безусловной оптимизации:** Задача безусловной оптимизации.

**Одномерной или многомерной оптимизации:** Многомерной: (двумерной).

**Функция унимодальная или многоэкстремальная:** Функция многоэкстремальная.

**Функция стохастическая или нет:** Функция не стохастическая.

**Особенности:** Нет.

## 1.14.6 Реализация

Реализация функции взята из библиотеки HarrixMathLibrary в разделе «Тестовые функции для оптимизации», которую можно найти по адресу <https://github.com/Harrix/HarrixMathLibrary>.

Код 1.14. Код функции MHL\_TestFunction\_Rana

```
double MHL_TestFunction_Rana(double x, double y)
{
    /*
Функция двух переменных: функция Rana.
Тестовая функция вещественной оптимизации.
Входные параметры:
x - первая вещественная переменная;
y - вторая вещественная переменная.
Возвращаемое значение:
Значение тестовой функции в точке (x, y).
    */
```

```

*/
double VMHL_Result;
VMHL_Result=x*sin(sqrt(fabs(y+1.-x)))*cos(sqrt(fabs(y+1.+x))) + (y+1.)*cos(sqrt(fabs(
    y+1.-x)))*sin(sqrt(fabs(y+1.+x)));
return VMHL_Result;
}

```

## 1.14.7 Ссылки

Данная функция приводится в следующих источниках:

1. [15] – Rana.

## 1.15 Функция ReverseGriewank

### 1.15.1 Описание функции

**Идентификатор:**

MHL\_TestFunction\_ReverseGriewank.

**Наименование:**

Функция ReverseGriewank.

**Тип:**

Задача вещественной оптимизации.

**Формула** (целевая функция):

$$f(\bar{x}) = \frac{1}{\frac{\bar{x}_1^2 + \bar{x}_2^2}{200} - \cos(\bar{x}_0) \cos\left(\frac{\bar{x}_2}{\sqrt{2}}\right) + 2}, \text{ где } \quad (1.16)$$

$\bar{x} \in X$ ,  $\bar{x}_j \in [Left_j; Right_j]$ ,  $Left_j = -10$ ,  $Right_j = 10$ ,  $j = \overline{1, n}$ ,  $n = 2$ .

**Обозначение:**

$\bar{x}$  — вещественный вектор;

$n = 2$  — размерность вещественного вектора.

**Решаемая задача оптимизации:**

$\bar{x}_{max} = \arg \max_{\bar{x} \in X} f(\bar{x})$ .

**Точка максимума:**

$\bar{x}_{max} = (0, 0)^T$ , то есть  $(\bar{x}_{max})_j = 0$  ( $j = \overline{1, n}$ ).

**Максимум функции:**

$f(\bar{x}_{max}) = 1$ .

**График:**

Рисунок 1.16 на с. 55 стр.

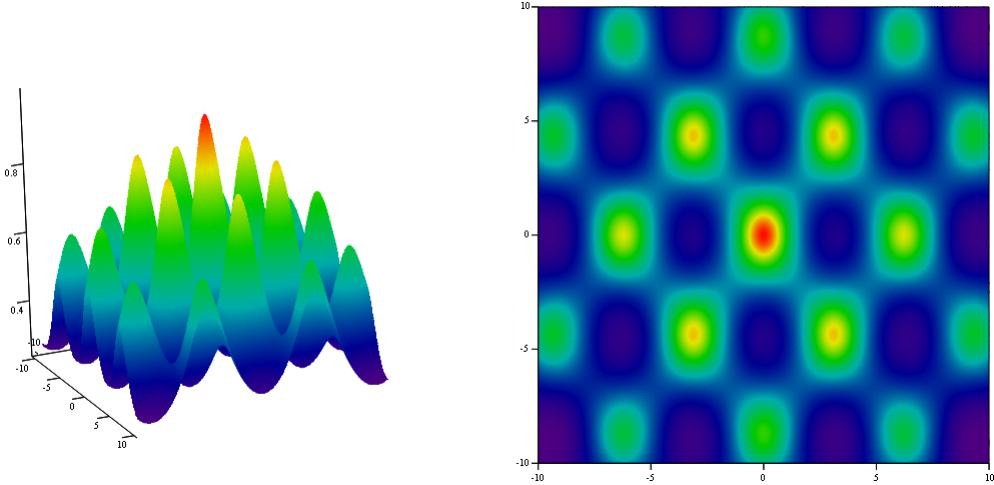


Рисунок 1.16. Функция ReverseGriewank

### 1.15.2 Параметры для алгоритмов оптимизации

**Точность вычислений:**

$$\varepsilon = 0.05.$$

**Число интервалов, на которые предполагается разбивать каждую компоненту вектора  $\bar{x}$  в пределах своего изменения** (для алгоритмов дискретной оптимизации) :

**Для этого длина бинарной строки для  $x_j$  координаты равна** (для алгоритмов бинарной оптимизации) :

$$Number\text{Of} \text{ } Parts_j = 4095 \ (j = \overline{1, n}).$$

**Замечание:**  $Number\text{Of} \text{ } Parts_j$  выбирается как минимальное число, удовлетворяющее соотношению:

$$Number\text{Of} \text{ } Parts_j = 2^{(k_2)_j} - 1 \geq \frac{10 (Right_j - Left_j)}{\varepsilon}, \text{ где } (k_2)_j \in \mathbb{N}, (j = \overline{1, n}).$$

### 1.15.3 Основная задача и подзадачи

**Изменяемый параметр:**

$n$  — размерность вещественного вектора.

**Значение в основной задаче:**

$$n = 2.$$

### 1.15.4 Нахождение ошибки оптимизации

Пусть в результате работы алгоритма оптимизации за  $N$  запусков мы нашли решения  $\bar{x}_{submin}^k$  со значениями целевой функции  $f(\bar{x}_{submin}^k)$  соответственно ( $k = \overline{1, N}$ ). Используем три вида ошибок:

**Надёжность:**

$$R = \frac{\sum_{k=1}^N S(\bar{x}_{submin}^k)}{N}, \text{ где}$$
$$S(\bar{x}_{submin}^k) = \begin{cases} 1, & \text{если } \left| (\bar{x}_{submin}^k)_j - (\bar{x}_{min})_j \right| < \varepsilon, j = \overline{1, n}; \\ 0, & \text{иначе.} \end{cases}$$

**Ошибка по входным параметрам:**

$$E_x = \frac{\sum_{k=1}^N \left( \frac{\sqrt{\sum_{j=1}^n ((\bar{x}_{submin}^k)_j - (\bar{x}_{min})_j)^2}}{n} \right)}{N}.$$

**Ошибка по значениям целевой функции:**

$$E_f = \frac{\sum_{k=1}^N |f(\bar{x}_{submin}^k) - f(\bar{x}_{min})|}{N}.$$

## 1.15.5 Свойства задачи

**Условной или безусловной оптимизации:** Задача безусловной оптимизации.

**Одномерной или многомерной оптимизации:** Многомерной: (двумерной).

**Функция унимодальная или многоэкстремальная:** Функция многоэкстремальная.

**Функция стохастическая или нет:** Функция не стохастическая.

**Особенности:** Нет.

## 1.15.6 Реализация

Реализация функции взята из библиотеки HarrixMathLibrary в разделе «Тестовые функции для оптимизации», которую можно найти по адресу <https://github.com/Harrix/HarrixMathLibrary>.

Код 1.15. Код функции MHL\_TestFunction\_ReverseGriewank

```
double MHL_TestFunction_ReverseGriewank(double x, double y)
{
    /*

Функция двух переменных: функция ReverseGriewank.

Тестовая функция вещественной оптимизации.

Входные параметры:
    x - первая вещественная переменная;
    y - вторая вещественная переменная.

Возвращаемое значение:
    Значение тестовой функции в точке (x, y).
}
```

```

*/
double VMHL_Result;

VMHL_Result = 1./((x*x+y*y)/200.-cos(x)*cos(y/sqrt(2.))+2.);

return VMHL_Result;
}

```

## 1.15.7 Ссылки

Так и не смог найти нормальный источник для этой функции. По внешнему виду похожа на функцию Грибанка, которую возвели в  $-1$  степень. Откуда-то у меня находится со студенческих времен.

## 1.16 Функция Сомбреро

### 1.16.1 Описание функции

**Идентификатор:**

MHL\_TestFunction\_Sombrero.

**Наименование:**

Функция Сомбреро.

**Тип:**

Задача вещественной оптимизации.

**Формула** (целевая функция):

$$f(\bar{x}) = \frac{1 - \sin\left(\sqrt{\bar{x}_1^2 + \bar{x}_2^2}\right)^2}{1 + 0.001(\bar{x}_1^2 + \bar{x}_2^2)}, \text{ где} \quad (1.17)$$

$\bar{x} \in X$ ,  $\bar{x}_j \in [Left_j; Right_j]$ ,  $Left_j = -10$ ,  $Right_j = 10$ ,  $j = \overline{1, n}$ ,  $n = 2$ .

**Обозначение:**

$\bar{x}$  — вещественный вектор;

$n = 2$  — размерность вещественного вектора.

**Решаемая задача оптимизации:**

$\bar{x}_{max} = \arg \max_{\bar{x} \in X} f(\bar{x})$ .

**Точка максимума:**

$\bar{x}_{max} = (0, 0)^T$ , то есть  $(\bar{x}_{max})_j = 0$  ( $j = \overline{1, n}$ ).

**Максимум функции:**

$f(\bar{x}_{max}) = 1$ .

**График:**

Рисунок 1.17 на с. 58 стр.

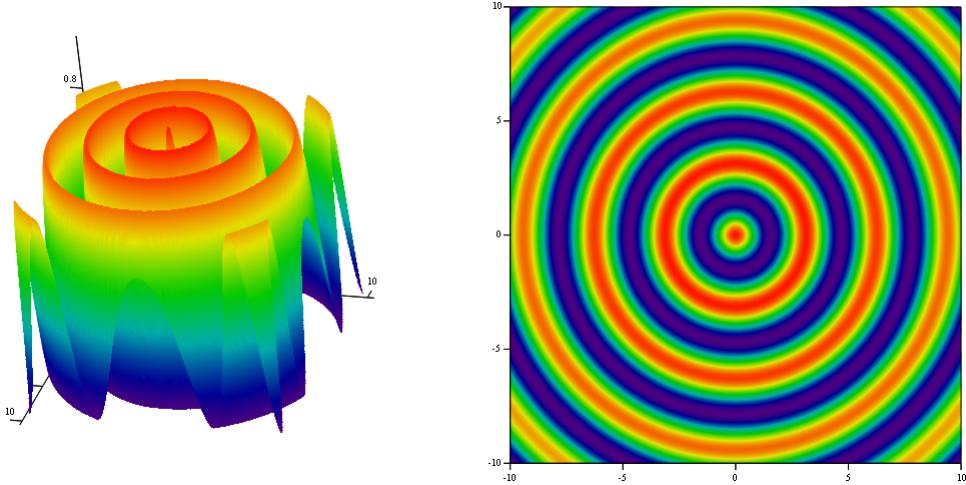


Рисунок 1.17. Функция Сомбреро

## 1.16.2 Параметры для алгоритмов оптимизации

**Точность вычислений:**

$$\varepsilon = 0.05.$$

**Число интервалов, на которые предполагается разбивать каждую компоненту вектора  $\bar{x}$  в пределах своего изменения** (для алгоритмов дискретной оптимизации) :

**Для этого длина бинарной строки для  $x_j$  координаты равна** (для алгоритмов бинарной оптимизации) :

$$(k_2)_j = 12 \quad (j = \overline{1, n}).$$

**Замечание:**  $NumberOfParts_j$  выбирается как минимальное число, удовлетворяющее соотношению:

$$NumberOfParts_j = 2^{(k_2)_j} - 1 \geq \frac{10 (Right_j - Left_j)}{\varepsilon}, \text{ где } (k_2)_j \in \mathbb{N}, (j = \overline{1, n}).$$

## 1.16.3 Основная задача и подзадачи

**Изменяемый параметр:**

$n$  — размерность вещественного вектора.

**Значение в основной задаче:**

$$n = 2.$$

## 1.16.4 Нахождение ошибки оптимизации

Пусть в результате работы алгоритма оптимизации за  $N$  запусков мы нашли решения  $\bar{x}_{submin}^k$  со значениями целевой функции  $f(\bar{x}_{submin}^k)$  соответственно ( $k = \overline{1, N}$ ). Используем три вида ошибок:

**Надёжность:**

$$R = \frac{\sum_{k=1}^N S(\bar{x}_{submin}^k)}{N}, \text{ где}$$
$$S(\bar{x}_{submin}^k) = \begin{cases} 1, & \text{если } \left| (\bar{x}_{submin}^k)_j - (\bar{x}_{min})_j \right| < \varepsilon, j = \overline{1, n}; \\ 0, & \text{иначе.} \end{cases}$$

**Ошибка по входным параметрам:**

$$E_x = \frac{\sum_{k=1}^N \left( \frac{\sqrt{\sum_{j=1}^n ((\bar{x}_{submin}^k)_j - (\bar{x}_{min})_j)^2}}{n} \right)}{N}.$$

**Ошибка по значениям целевой функции:**

$$E_f = \frac{\sum_{k=1}^N |f(\bar{x}_{submin}^k) - f(\bar{x}_{min})|}{N}.$$

## 1.16.5 Свойства задачи

**Условной или безусловной оптимизации:** Задача безусловной оптимизации.

**Одномерной или многомерной оптимизации:** Многомерной: (двумерной).

**Функция унимодальная или многоэкстремальная:** Функция многоэкстремальная.

**Функция стохастическая или нет:** Функция не стохастическая.

**Особенности:** Нет.

## 1.16.6 Реализация

Реализация функции взята из библиотеки HarrixMathLibrary в разделе «Тестовые функции для оптимизации», которую можно найти по адресу <https://github.com/Harrix/HarrixMathLibrary>.

Код 1.16. Код функции MHL\_TestFunction\_Sombrero

```
double MHL_TestFunction_Sombrero(double x, double y)
```

```
{  
/*
```

*Функция одной переменных: функция Сомбреро.*

*Тестовая функция вещественной оптимизации.*

*Входные параметры:*

*x – первая вещественная переменная;*

*y – вторая вещественная переменная.*

*Возвращаемое значение:*

*Значение тестовой функции в точке (x, y).*

```

*/
double VMHL_Result;
VMHL_Result = 1.-sin(sqrt(x*x+y*y))*sin(sqrt(x*x+y*y));
VMHL_Result /= (1.+0.001*(x*x+y*y));
return VMHL_Result;
}

```

## 1.16.7 Ссылки

Данная функция приводится в следующих источниках:

1. [12, стр. 30] — Эволюционные методы моделирования и оптимизации сложных систем.

## 1.17 Функция Multiextremal

### 1.17.1 Описание функции

**Идентификатор:**

MHL\_TestFunction\_Multiextremal.

**Наименование:**

Multiextremal.

**Тип:**

Задача вещественной оптимизации.

**Формула** (целевая функция):

$$f(\bar{x}) = 0.05(x-1)^2 + \left(3 - 2.9e^{-2.77257x^2}\right) \left(1 - \cos\left(x\left(4 - 50e^{-2.77257x^2}\right)\right)\right), \text{ где} \quad (1.18)$$

$\bar{x} \in X$ ,  $\bar{x}_j \in [Left_j; Right_j]$ ,  $Left_j = -2$ ,  $Right_j = 2$ ,  $j = \overline{1, n}$ ,  $n = 1$ .

**Обозначение:**

$\bar{x}$  — вещественный вектор;

$n = 1$  — размерность вещественного вектора.

**Решаемая задача оптимизации:**

$\bar{x}_{min} = \arg \min_{\bar{x} \in X} f(\bar{x})$ .

**Точка минимума:**

$\bar{x}_{min} \approx (0.954452)^T$ , то есть  $(\bar{x}_{min})_j \approx 0.954452 (j = \overline{1, n})$ .

**Минимум функции:**

$f(\bar{x}_{min}) \approx 0.000103742$ .

**График:**

Рисунок 1.18 на с. 61

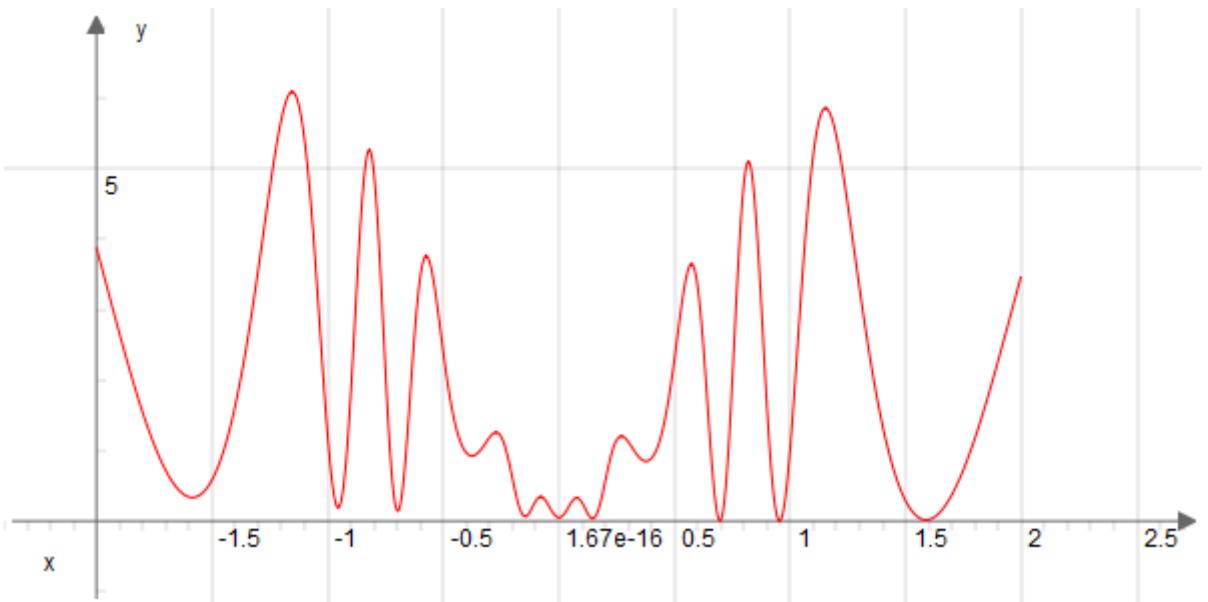


Рисунок 1.18. Функция Multiextremal

### 1.17.2 Параметры для алгоритмов оптимизации

**Точность вычислений:**

$$\varepsilon = 0.01.$$

**Число интервалов, на которые предполагается разбивать каждую компоненту вектора  $\bar{x}$  в пределах своего изменения** (для алгоритмов дискретной оптимизации) :

**Для этого длина бинарной строки для  $x_j$  координаты равна** (для алгоритмов бинарной оптимизации) :

**Замечание:**  $NumberOfParts_j$  выбирается как минимальное число, удовлетворяющее соотношению:

$$NumberOfParts_j = 2^{(k_2)_j} - 1 \geq \frac{10(Right_j - Left_j)}{\varepsilon}, \text{ где } (k_2)_j \in \mathbb{N}, (j = \overline{1, n}).$$

### 1.17.3 Основная задача и подзадачи

**Изменяемый параметр:**

$n$  — размерность вещественного вектора.

**Значение в основной задаче:**

$$n = 1.$$

#### 1.17.4 Нахождение ошибки оптимизации

Пусть в результате работы алгоритма оптимизации за  $N$  запусков мы нашли решения  $\bar{x}_{submin}^k$  со значениями целевой функции  $f(\bar{x}_{submin}^k)$  соответственно ( $k = \overline{1, N}$ ). Используем три вида ошибок:

**Надёжность:**

$$R = \frac{\sum_{k=1}^N S(\bar{x}_{submin}^k)}{N}, \text{ где}$$
$$S(\bar{x}_{submin}^k) = \begin{cases} 1, & \text{если } |(\bar{x}_{submin}^k)_j - (\bar{x}_{min})_j| < \varepsilon, j = \overline{1, n}; \\ 0, & \text{иначе.} \end{cases}$$

**Ошибка по входным параметрам:**

$$E_x = \frac{\sum_{k=1}^N \left( \frac{\sqrt{\sum_{j=1}^n ((\bar{x}_{submin}^k)_j - (\bar{x}_{min})_j)^2}}{n} \right)}{N}.$$

**Ошибка по значениям целевой функции:**

$$E_f = \frac{\sum_{k=1}^N |f(\bar{x}_{submin}^k) - f(\bar{x}_{min})|}{N}.$$

#### 1.17.5 Свойства задачи

**Условной или безусловной оптимизации:** Задача безусловной оптимизации.

**Одномерной или многомерной оптимизации:** Одномерной.

**Функция унимодальная или многоэкстремальная:** Функция многоэкстремальная.

**Функция стохастическая или нет:** Функция не стохастическая.

**Особенности:** Нет.

#### 1.17.6 Реализация

Реализация функции взята из библиотеки HarrixMathLibrary в разделе «Тестовые функции для оптимизации», которую можно найти по адресу <https://github.com/Harrix/HarrixMathLibrary>.

Код 1.17. Код функции MHL\_TestFunction\_Multiextremal

```
double MHL_TestFunction_Multiextremal(double x)
{
    /*
```

```

Функция одной переменных: функция Multiextremal.
Тестовая функция вещественной оптимизации.
Входные параметры:
    x - вещественная переменная.
Возвращаемое значение:
    Значение тестовой функции в точке (x).
*/
double VMHL_Result;
VMHL_Result = (0.05*(x-1.)*(x-1.) + (3.-2.9*exp(-2.77257*x*x))* (1-cos(x*(4.-50*exp
(-2.77257*x*x)))) );
return VMHL_Result;
}

```

## 1.17.7 Ссылки

Данная функция приводится в следующих источниках:

1. [12, стр. 26] — Эволюционные методы моделирования и оптимизации сложных систем.

По сравнению с данной работой в данном документе представлено уточненное значение функции в точке минимума.

## 1.18 Функция Multiextremal2

### 1.18.1 Описание функции

**Идентификатор:**

MHL\_TestFunction\_Multiextremal2.

**Наименование:**

Multiextremal2.

**Тип:**

Задача вещественной оптимизации.

**Формула** (целевая функция):

$$f(\bar{x}) = 1 - 0.5 \cos(1.5(10x - 0.3)) \cos(31.4x) + 0.5 \cos(\sqrt{5} \cdot 10x) \cos(35x), \text{ где } \quad (1.19)$$

$\bar{x} \in X$ ,  $\bar{x}_j \in [Left_j; Right_j]$ ,  $Left_j = -2$ ,  $Right_j = 2$ ,  $j = \overline{1, n}$ ,  $n = 1$ .

**Обозначение:**

$\bar{x}$  — вещественный вектор;

$n = 1$  — размерность вещественного вектора.

**Решаемая задача оптимизации:**

$\bar{x}_{max} = \arg \max_{\bar{x} \in X} f(\bar{x})$ .

**Точка максимума:**

$\bar{x}_{max} \approx (-0.993263)^T$ , то есть  $(\bar{x}_{max})_j \approx -0.993263$  ( $j = \overline{1, n}$ ).

**Максимум функции:**

$f(\bar{x}_{max}) \approx 1.93374$ .

**График:**

Рисунок 1.19 на 64 стр.

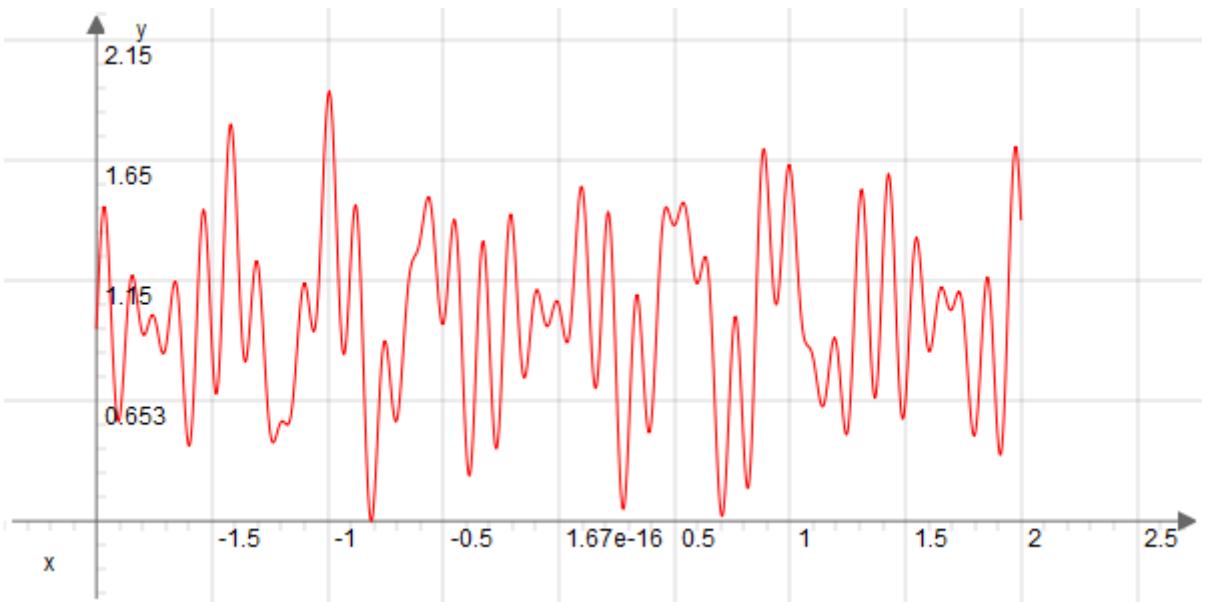


Рисунок 1.19. Функция Multiextremal2

### 1.18.2 Параметры для алгоритмов оптимизации

**Точность вычислений:**

$$\varepsilon = 0.01.$$

**Число интервалов, на которые предполагается разбивать каждую компоненту вектора  $\bar{x}$  в пределах своего изменения** (для алгоритмов дискретной оптимизации) :

**Для этого длина бинарной строки для  $x_j$  координаты равна** (для алгоритмов бинарной оптимизации) :

**Замечание:**  $NumberOfParts_j$  выбирается как минимальное число, удовлетворяющее соотношению:

$$NumberOfParts_j = 2^{(k_2)_j} - 1 \geq \frac{10 (Right_j - Left_j)}{\varepsilon}, \text{ где } (k_2)_j \in \mathbb{N}, (j = \overline{1, n}).$$

### 1.18.3 Основная задача и подзадачи

**Изменяемый параметр:**

$n$  — размерность вещественного вектора.

**Значение в основной задаче:**

$$n = 1.$$

#### 1.18.4 Нахождение ошибки оптимизации

Пусть в результате работы алгоритма оптимизации за  $N$  запусков мы нашли решения  $\bar{x}_{submin}^k$  со значениями целевой функции  $f(\bar{x}_{submin}^k)$  соответственно ( $k = \overline{1, N}$ ). Используем три вида ошибок:

**Надёжность:**

$$R = \frac{\sum_{k=1}^N S(\bar{x}_{submin}^k)}{N}, \text{ где}$$
$$S(\bar{x}_{submin}^k) = \begin{cases} 1, & \text{если } |(\bar{x}_{submin}^k)_j - (\bar{x}_{min})_j| < \varepsilon, j = \overline{1, n}; \\ 0, & \text{иначе.} \end{cases}$$

**Ошибка по входным параметрам:**

$$E_x = \frac{\sum_{k=1}^N \left( \frac{\sqrt{\sum_{j=1}^n ((\bar{x}_{submin}^k)_j - (\bar{x}_{min})_j)^2}}{n} \right)}{N}.$$

**Ошибка по значениям целевой функции:**

$$E_f = \frac{\sum_{k=1}^N |f(\bar{x}_{submin}^k) - f(\bar{x}_{min})|}{N}.$$

#### 1.18.5 Свойства задачи

**Условной или безусловной оптимизации:** Задача безусловной оптимизации.

**Одномерной или многомерной оптимизации:** Одномерной.

**Функция унимодальная или многоэкстремальная:** Функция многоэкстремальная.

**Функция стохастическая или нет:** Функция не стохастическая.

**Особенности:** Нет.

#### 1.18.6 Реализация

Реализация функции взята из библиотеки HarrixMathLibrary в разделе «Тестовые функции для оптимизации», которую можно найти по адресу <https://github.com/Harrix/HarrixMathLibrary>.

Код 1.18. Код функции MHL\_TestFunction\_Multiextremal2

```
double MHL_TestFunction_Multiextremal2(double x)
{
    /*
```

```

Функция одной переменных: функция Multiextremal2.
Тестовая функция вещественной оптимизации.
Входные параметры:
    x - вещественная переменная.
Возвращаемое значение:
    Значение тестовой функции в точке (x).
*/
double VMHL_Result;
VMHL_Result = 1.-0.5*cos(1.5*(10.*x-0.3))*cos(31.4*x)+0.5*cos(sqrt(5.)*10.*x)*cos
(35.*x);
return VMHL_Result;
}

```

## 1.18.7 Ссылки

Данная функция приводится в следующих источниках:

1. [12, стр. 27] — Эволюционные методы моделирования и оптимизации сложных систем.

По сравнению с данной работой в данном документе представлено уточненное значение функции в точке минимума.

## 1.19 Функция волна

### 1.19.1 Описание функции

**Идентификатор:**

MHL\_TestFunction\_Wave.

**Наименование:**

Волна.

**Тип:**

Задача вещественной оптимизации.

**Формула** (целевая функция):

$$f(\bar{x}) = e^{-\bar{x}_1^2} + 0.01 \cos(200 \cdot \bar{x}_1), \text{ где} \quad (1.20)$$

$\bar{x} \in X, \bar{x}_j \in [Left_j; Right_j], Left_j = -2, Right_j = 2, j = \overline{1, n}, n = 1.$

**Обозначение:**

$\bar{x}$  — вещественный вектор;

$n = 1$  — размерность вещественного вектора.

**Решаемая задача оптимизации:**

$\bar{x}_{max} = \arg \max_{\bar{x} \in X} f(\bar{x}).$

**Точка максимума:**

$\bar{x}_{max} = (0)^T$ , то есть  $(\bar{x}_{max})_j = 0 (j = \overline{1, n}).$

**Максимум функции:**

$f(\bar{x}_{max}) = 1.01.$

**График:**

Рисунок 1.20 на 67 стр.

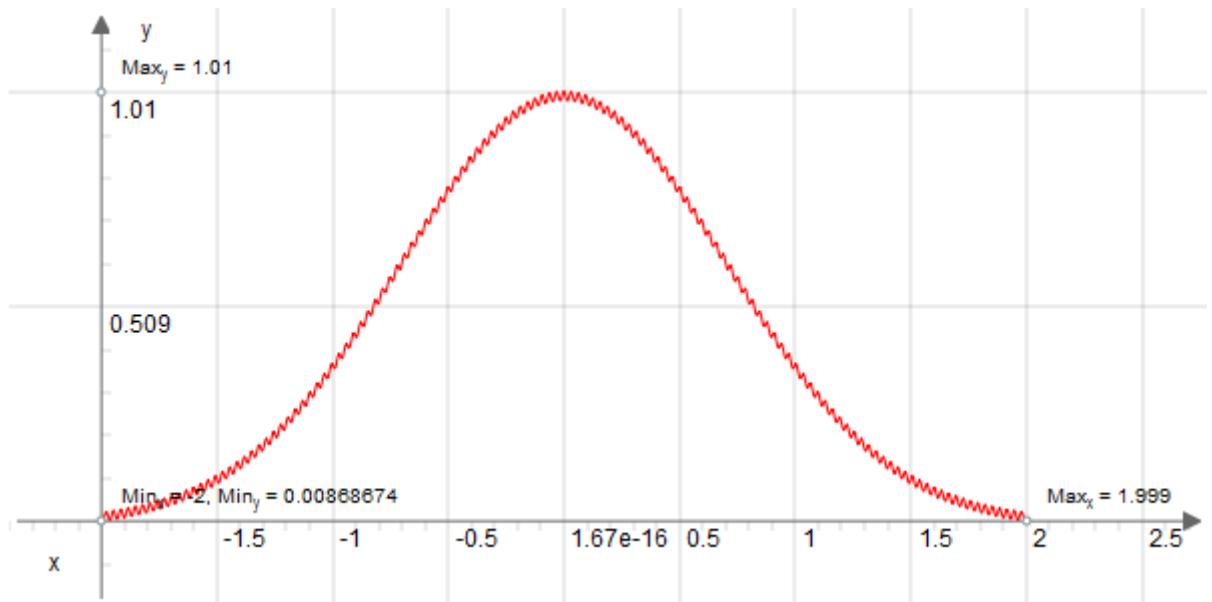


Рисунок 1.20. Волна

### 1.19.2 Параметры для алгоритмов оптимизации

**Точность вычислений:**

$$\varepsilon = 0.01.$$

**Число интервалов, на которые предполагается разбивать каждую компоненту вектора  $\bar{x}$  в пределах своего изменения** (для алгоритмов дискретной оптимизации) :

**Для этого длина бинарной строки для  $x_j$  координаты равна** (для алгоритмов бинарной оптимизации) :

**Замечание:**  $NumberOfParts_j$  выбирается как минимальное число, удовлетворяющее соотношению:

$$NumberOfParts_j = 2^{(k_2)_j} - 1 \geq \frac{10 (Right_j - Left_j)}{\varepsilon}, \text{ где } (k_2)_j \in \mathbb{N}, (j = \overline{1, n}).$$

### 1.19.3 Основная задача и подзадачи

**Изменяемый параметр:**

$n$  — размерность вещественного вектора.

**Значение в основной задаче:**

$$n = 1.$$

## 1.19.4 Нахождение ошибки оптимизации

Пусть в результате работы алгоритма оптимизации за  $N$  запусков мы нашли решения  $\bar{x}_{submin}^k$  со значениями целевой функции  $f(\bar{x}_{submin}^k)$  соответственно ( $k = \overline{1, N}$ ). Используем три вида ошибок:

**Надёжность:**

$$R = \frac{\sum_{k=1}^N S(\bar{x}_{submin}^k)}{N}, \text{ где}$$
$$S(\bar{x}_{submin}^k) = \begin{cases} 1, & \text{если } |(\bar{x}_{submin}^k)_j - (\bar{x}_{min})_j| < \varepsilon, j = \overline{1, n}; \\ 0, & \text{иначе.} \end{cases}$$

**Ошибка по входным параметрам:**

$$E_x = \frac{\sum_{k=1}^N \left( \frac{\sqrt{\sum_{j=1}^n ((\bar{x}_{submin}^k)_j - (\bar{x}_{min})_j)^2}}{n} \right)}{N}.$$

**Ошибка по значениям целевой функции:**

$$E_f = \frac{\sum_{k=1}^N |f(\bar{x}_{submin}^k) - f(\bar{x}_{min})|}{N}.$$

## 1.19.5 Свойства задачи

**Условной или безусловной оптимизации:** Задача безусловной оптимизации.

**Одномерной или многомерной оптимизации:** Одномерной.

**Функция унимодальная или многоэкстремальная:** Функция многоэкстремальная.

**Функция стохастическая или нет:** Функция не стохастическая.

**Особенности:** Хотя внешне можно отнести эту функцию к стохастической, так как по поведению напоминает вид плотности нормального распределения с помехой.

## 1.19.6 Реализация

Реализация функции взята из библиотеки HarrixMathLibrary в разделе «Тестовые функции для оптимизации», которую можно найти по адресу <https://github.com/Harrix/HarrixMathLibrary>.

Код 1.19. Код функции MHL\_TestFunction\_Wave

```
double MHL_TestFunction_Wave(double x)
```

```
{  
/*  
Функция одной переменных: волна.  
Тестовая функция вещественной оптимизации.  
Входные параметры:  
    x - вещественная переменная.  
Возвращаемое значение:  
    Значение тестовой функции в точке (x).  
*/  
double VMHL_Result;  
VMHL_Result = (exp(-x*x)+0.01*cos(200*x));  
return VMHL_Result;  
}
```

## 1.19.7 Ссылки

Так и не смог найти нормальный источник для этой функции. Откуда-то у меня находится со студенческих времен.

# Глава 2

## Задачи бинарной оптимизации

### 2.1 Сумма всех элементов бинарного вектора

#### 2.1.1 Описание функции

**Идентификатор:**

MHL\_TestFunction\_SumVector.

**Наименование:**

Сумма всех элементов бинарного вектора.

**Тип:**

Задача бинарной оптимизации.

**Формула** (целевая функция):

$$f(\bar{x}) = \sum_{i=1}^n \bar{x}_i, \text{ где } \quad (2.1)$$

$\bar{x} \in X, \bar{x}_j \in \{0; 1\}, j = \overline{1, n}.$

**Обозначение:**

$\bar{x}$  — бинарный вектор;

$n$  — размерность бинарного вектора.

**Объем поискового пространства:**

$\mu(X) = 2^n.$

**Решаемая задача оптимизации:**

$\bar{x}_{max} = \arg \max_{\bar{x} \in X} f(\bar{x}).$

**Точка максимума:**

$\bar{x}_{max} = (1, 1, \dots, 1)^T$ , то есть  $(\bar{x}_{max})_j = 1$  ( $j = \overline{1, n}$ ).

**Максимум функции:**

$f(\bar{x}_{max}) = n.$

**Точка минимума:**

$\bar{x}_{min} = (0, 0, \dots, 0)^T$ , то есть  $(\bar{x}_{min})_j = 0$  ( $j = \overline{1, n}$ ).

**Минимум функции:**

$f(\bar{x}_{min}) = 0.$

## 2.1.2 Основная задача и подзадачи

<b>Изменяемый параметр:</b>	$n$ — размерность бинарного вектора.
<b>Значение в основной задаче:</b>	$n = 20$ .
<b>Подзадача №2:</b>	$n = 30$ .
<b>Подзадача №3:</b>	$n = 40$ .
<b>Подзадача №4:</b>	$n = 50$ .
<b>Подзадача №5:</b>	$n = 60$ .
<b>Подзадача №6:</b>	$n = 70$ .
<b>Подзадача №7:</b>	$n = 80$ .
<b>Подзадача №8:</b>	$n = 90$ .
<b>Подзадача №9:</b>	$n = 100$ .
<b>Подзадача №10:</b>	$n = 200$ .

## 2.1.3 Нахождение ошибки оптимизации

Пусть в результате работы алгоритма оптимизации за  $N$  запусков мы нашли решения  $\bar{x}_{submax}^k$  со значениями целевой функции  $f(\bar{x}_{submax}^k)$  соответственно ( $k = \overline{1, N}$ ). Используем три вида ошибок:

**Надёжность:**

$$R = \frac{\sum_{k=1}^N S(\bar{x}_{submax}^k)}{N}, \text{ где}$$

$$S(\bar{x}_{submax}^k) = \begin{cases} 1, & \text{если } \bar{x}_{submax}^k = \bar{x}_{max}; \\ 0, & \text{иначе.} \end{cases}$$

**Ошибка по входным параметрам:**

$$E_x = \frac{\sum_{k=1}^N \left( \frac{\sum_{j=1}^n |(\bar{x}_{submax}^k)_j - (\bar{x}_{max})_j|}{n} \right)}{N}.$$

**Ошибка по значениям целевой функции:**

$$E_f = \frac{\sum_{k=1}^N \left( \frac{|f(\bar{x}_{submax}^k) - f(\bar{x}_{max})|}{n} \right)}{N}.$$

## 2.1.4 Свойства задачи

**Условной или безусловной оптимизации:** Задача безусловной оптимизации.

**Одномерной или многомерной оптимизации:** Многомерной:  $n$ .

**Функция унимодальная или многоэкстремальная:** Функция унимодальная.

**Функция стохастическая или нет:** Функция не стохастическая.

**Особенности:** Нет.

## 2.1.5 Реализация

Реализация функции взята из библиотеки HarrixMathLibrary в разделе «Тестовые функции для оптимизации», которую можно найти по адресу <https://github.com/Harrix/HarrixMathLibrary>.

Код 2.1. Код функции MHL\_TestFunction\_SumVector

```
double MHL_TestFunction_SumVector(int *x, int VMHL_N)
{
/*
Сумма всех элементов бинарного вектора.
Тестовая функция бинарной оптимизации.
Входные параметры:
    x - указатель на исходный массив;
    VMHL_N - размер массива x.
Возвращаемое значение:
    Значение тестовой функции в точке x.
*/
double VMHL_Result=0;
for (int i=0;i<VMHL_N;i++) VMHL_Result+=x[i];
return VMHL_Result;
}
```

# Литература

1. Dieterich Johannes M., Hartke Bernd. Empirical review of standard benchmark functions using evolutionary global optimization // CoRR. 2012. Т. abs/1207.4318. <http://arxiv.org/pdf/1207.4318v1.pdf>.
2. Ackley's Function. <http://www.cs.unm.edu/~neal.holts/dga/benchmarkFunction/ackley.html>.
3. Paraboloid. <http://en.wikipedia.org/wiki/Paraboloid>.
4. Rastrigin function. [http://en.wikipedia.org/wiki/Rastrigin\\_function](http://en.wikipedia.org/wiki/Rastrigin_function).
5. Non-linear Continuous Multi-Extremal Optimization. <http://www.maths.uq.edu.au/CEToolBox/node3.html>.
6. Parametric Optimization. <http://www.pg.gda.pl/~mkwies/dyd/gedocu/fcnfun6.html>.
7. Rosenbrock function. [http://en.wikipedia.org/wiki/Rosenbrock\\_function](http://en.wikipedia.org/wiki/Rosenbrock_function).
8. Rosenbrock Function. [http://www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar\\_files/TestGO\\_files/Page2537.htm](http://www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar_files/TestGO_files/Page2537.htm).
9. Pohlheim Hartmut. GEATbx Examples. Examples of Objective Functions. 2006. [http://www.geatbx.com/download/GEATbx\\_ObjFunExpl\\_v38.pdf](http://www.geatbx.com/download/GEATbx_ObjFunExpl_v38.pdf).
10. Virtual Library of Simulation Experiments: Test Functions and Datasets. Rotated hyper-ellipsoid function. 2013. <http://www.sfu.ca/~ssurjano/rothyp.html>.
11. Huang D.S., Li K., Irwin G.W. International Conference on Intelligent Computing: Intelligent computing. International Conference on Intelligent Computing: ICIC 2006, Kunming, China, August 16-19, 2006 : Proceedings. Springer, 2006. <http://books.google.ru/books?id=7sH4RsXYu7cC>.
12. Эволюционные методы моделирования и оптимизации сложных систем / Е. С. Семенкин, М. Н. Жукова, В. Г. Жуков [и др.]. Красноярск: Федеральное агентство по образованию, Сибирский федеральный университет, 2007. 310 с. [http://files.lib.sfu-kras.ru/ebibl/umkd/22/u\\_lectures.pdf](http://files.lib.sfu-kras.ru/ebibl/umkd/22/u_lectures.pdf).
13. Himmelblau's function. [http://en.wikipedia.org/wiki/Himmelblau%27s\\_function](http://en.wikipedia.org/wiki/Himmelblau%27s_function).
14. Minimization of the Himmelblau Function. [http://pythonhosted.org/algopy/examples/minimization/himmelblau\\_minimization.html](http://pythonhosted.org/algopy/examples/minimization/himmelblau_minimization.html).
15. Rana. <http://www.cs.unm.edu/~neal.holts/dga/benchmarkFunction/rana.html>.