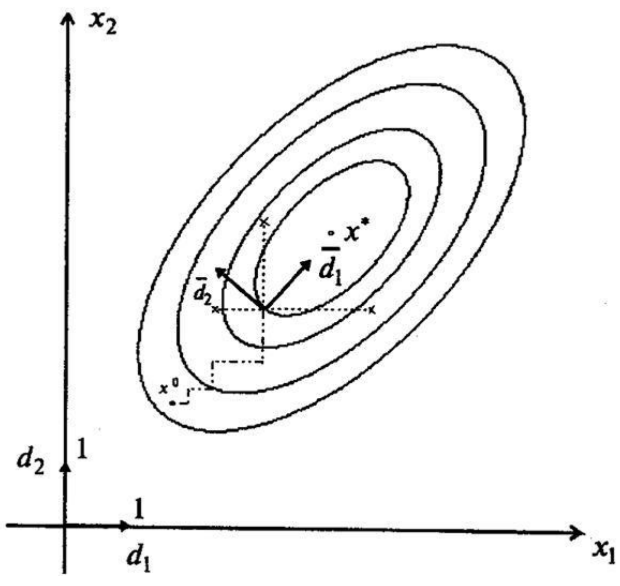
УНИВЕРСИТЕТ ИТМО

**С.В. Рыков, С.А. Рыков, И.В. Кудрявцева,
А.С. Старков, А.В. Свердлов**

МЕТОДЫ ОПТИМИЗАЦИИ В ПРИМЕРАХ В ПАКЕТЕ MATHCAD 15. ЧАСТЬ VI (ЧИСЛЕННЫЙ МЕТОД БЕЗУСЛОВНОЙ ОПТИМИЗАЦИИ НУЛЕВОГО ПРЯДКА. МЕТОД СЛУЧАЙНОГО ПОИСКА С ВОЗВРАТОМ ПРИ НЕУДАЧНОМ ШАГЕ)



**Санкт-Петербург
2020**

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

УНИВЕРСИТЕТ ИТМО

**С.В. Рыков, С.А. Рыков, И.В. Кудрявцева,
А.С. Старков, А.В. Свердлов**

МЕТОДЫ ОПТИМИЗАЦИИ В ПРИМЕРАХ В ПАКЕТЕ MATHCAD 15. ЧАСТЬ VI (ЧИСЛЕННЫЙ МЕТОД БЕЗУСЛОВНОЙ ОПТИМИЗАЦИИ НУЛЕВОГО ПРЯДКА. МЕТОД СЛУЧАЙНОГО ПОИСКА С ВОЗВРАТОМ ПРИ НЕУДАЧНОМ ШАГЕ)

УЧЕБНО-МЕТОДИЧЕСКОЕ ПОСОБИЕ

РЕКОМЕНДОВАНО К ИСПОЛЬЗОВАНИЮ В УНИВЕРСИТЕТЕ ИТМО
по направлению подготовки 16.04.03 Холодильная, криогенная техника и системы жизнеобеспечения в качестве учебно-методического пособия для реализации основных профессиональных образовательных программ высшего образования магистратуры

УНИВЕРСИТЕТ ИТМО

**Санкт-Петербург
2020**

Рыков С.В., Рыков С.А., Кудрявцева И.В., Старков А.С., Свердлов А.В., Методы оптимизации в примерах в пакете MATHCAD 15. Часть VI (Численный метод безусловной оптимизации нулевого прядка. Метод случайного поиска с возвратом при неудачном шаге)– СПб: Университет ИТМО, 2020. – 105 с.

Рецензент(ы):
Пронин Владимир Александрович, доктор технических наук, профессор, профессор (квалификационная категория " ординарный профессор") факультета низкотемпературной энергетики, Университета ИТМО.

Пособие содержит сведения о численных методах многомерной оптимизации – численном методе безусловной оптимизации нулевого прядка и методе случайного поиска с возвратом при неудачном шаге. Снабжено большим количеством примеров реализации оптимизационных задач, рассмотренных как численно при построчной реализации и с помощью единой пользовательской функции, так и с использованием функций пакета MathCAD15. Пособие также содержит примеры для самостоятельного решения и контрольные вопросы с пояснениями по рассмотренному методу. Предназначено для самостоятельной работы студентов и аспирантов ВУЗов очной и заочной форм обучения.

УНИВЕРСИТЕТ ИТМО

Университет ИТМО – ведущий вуз России в области информационных и фотонных технологий, один из немногих российских вузов, получивших в 2009 году статус национального исследовательского университета. С 2013 года Университет ИТМО – участник программы повышения конкурентоспособности российских университетов среди ведущих мировых научно-образовательных центров, известной как проект «5 в 100». Цель Университета ИТМО – становление исследовательского университета мирового уровня, предпринимательского по типу, ориентированного на интернационализацию всех направлений деятельности.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Цель настоящего пособия – оказать студентам конкретную помощь в развитии умения решать задачи курса «Методы оптимизации». Пособие содержит теоретические положения, подробное решение соответствующих задач как численным методом случайного поиска с возвратом при неудачном шаге, так и с использованием функций пакета MathCAD 15.

Первая часть пособия посвящена изложению вербальной модели и алгоритма поиска численным методом случайного поиска с возвратом при неудачном шаге. Вторая часть пособия содержит подробно разобранные примеры реализации алгоритма метода. Материал излагается с учетом терминологии и обозначений, предусмотренных программой ВУЗа.

При пользовании пособием рекомендуется следующий порядок работы. Сначала следует изучить теоретическую часть и проверить знания с помощью контрольных вопросов. Затем ознакомиться с пояснениями и подробными решениями задач, содержащимися в пособии при построчной реализации метода случайного поиска с возвратом при неудачном шаге и с использованием единой пользовательской функции. И только после этого перейти к выполнению контрольных упражнений, помещенных в конце пособия.

Методы нулевого порядка. Методы случайного поиска

Метод случайного поиска основан на применении последовательностей случайных чисел, с помощью которых в области изменения независимых переменных производится выборка случайных точек или определение случайных направлений. Этот метод является прямым развитием известного метода проб и ошибок, когда решение ищется случайно и при удаче принимается, а при неудаче отвергается с тем, чтобы немедленно снова обратиться к случайности как к источнику возможностей. Такое случайное поведение разумно опирается на уверенность, что случайность содержит в себе все возможности, в том числе и искомое решение во всех его вариантах.

Метод случайного поиска при оптимальном проектировании позволяет со сравнительно небольшими затратами машинного времени определить экстремум функции большого числа переменных. Достоинством этого метода является то, что, кроме необходимости существования в рассматриваемой области единственного локального экстремума, он не предъявляет существенных требований ни к виду множества параметров, по которым отыскивается оптимальное значение, ни к виду зависимостей, связывающих выбираемые параметры с оптимизирующим критерием и ограничениями. Он позволяет найти все локальные минимумы функции от 10–20 переменных со сложным рельефом. Он полезен и при исследовании функции с единственным минимумом.

Этот метод имеет два преимущества. Во-первых, он пригоден для любой целевой функции независимо от того, является она унимодальной или нет. Во-вторых, вероятность успеха при попытках не зависит от размерности рассматриваемого пространства. Хотя этот метод не позволяет непосредственно найти оптимальное решение, он создает подходящие предпосылки для применения в дальнейшем других методов поиска. Поэтому его часто применяют в сочетании с одним или несколькими методами других типов.

Недостаток метода в том, что надо заранее задать область, в которой выбираются случайные точки. Если мы зададим слишком широкую область, то ее труднее детально исследовать, а если выберем слишком узкую область, то многие локальные минимумы могут оказаться вне ее.

Существуют несколько методов случайного поиска, которые похожи друг на друга и отличаются всего несколькими шагами или условиями.

1 Метод случайного поиска с возвратом при неудачном шаге

1.1 Теория метода

Постановка задачи

Требуется найти безусловный минимум функции $f(x)$ многих переменных, то есть найти такую точку $x^* \in R^n$, что $f(x^*) = \min\limits_{x \in R^n} f(x)$.

1.1.1 Вербальная модель метода

Задается начальная точка x_0. Каждая последующая точка находится по формуле:

$$x_{k+1} = x_k + t_k \xi_k ,$$

где $t_k > 0$ – величина шага; ξ_k – случайный вектор единичной длины, определяющий направление поиска; k – номер итерации. На текущей итерации при помощи генерирования случайных векторов ξ_k получаются точки, лежащие на гиперсфере радиуса t_k с центром в точке x_k (см. Рис. 1.1).

Если значение функции в полученной точке не меньше, чем в центре, шаг считается неудачным, происходит возврат в текущий центр и поиск продолжается. Если число неудачных шагов из текущей точки достигает некоторого числа M, дальнейший поиск продолжается из той же точки, но с меньшим шагом до тех пор, пока он не станет меньше заранее заданной величины R. Если же значение функции в полученной точке меньше, чем в центре, шаг считается удачным, и центральная точка переносится в новую точку, и дальнейший поиск продолжается из этой точки.

1.1.2 Алгоритм поиска

Шаг 1. Задать начальную точку x_0, коэффициент сжатия β ($0 < \beta < 1$), M – максимальное число неудачно выполненных испытаний на текущей итерации, t_0 – начальную величину шага, R – минимальную величину шага, N – максимальное число итераций. Положить $\kappa = 0$, $j = 1$.

Шаг 2. Получить случайный вектор $\varsigma_j = (\varsigma_{j_1}, \ldots, \varsigma_{j_n})^T$, где ς_{j_i} – случайная величина, равномерно распределенная на интервале $[-1,1]$, n – количество элементов в векторе.

Шаг 3. Вычислить $y_j = x_k + t_k \xi_k$, где $\xi_j = \dfrac{\varsigma_j}{\|\varsigma_j\|}$ – случайный вектор единичной длины, $\|\xi\| = 1$, определяющий направление поиска, $\|\varsigma_j\|$ – модуль случайного вектора.

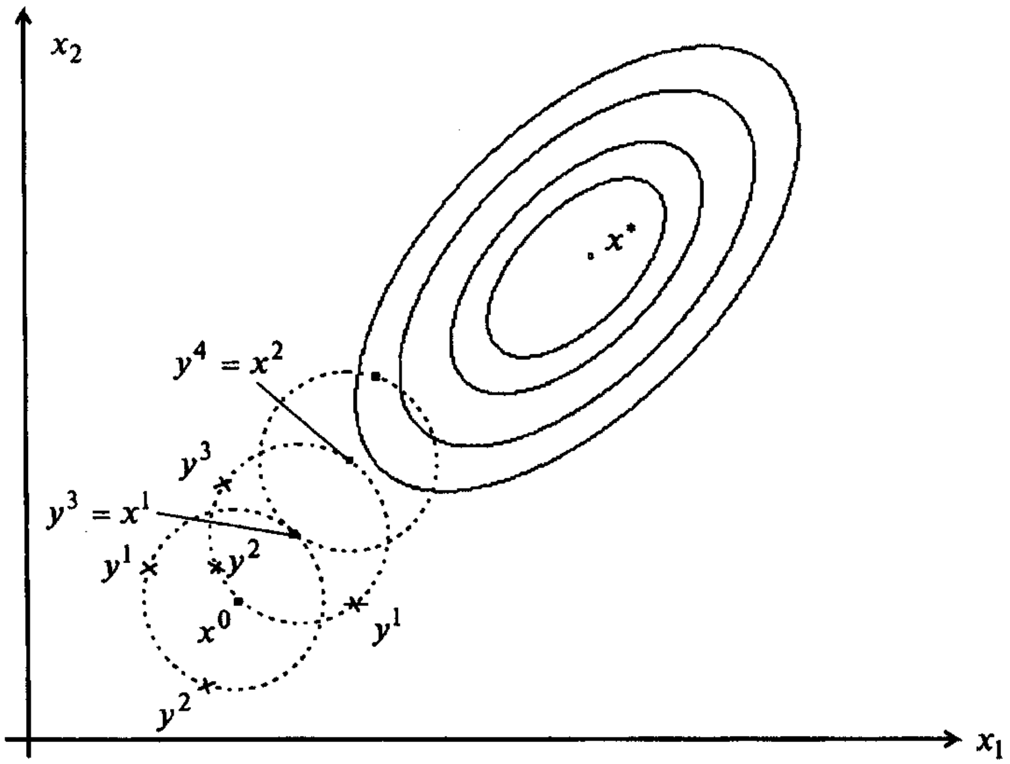


Рис. 1.1. Графическая модель реализации алгоритма метода случайного поиска с возвратом при неудачном шаге

Шаг 4. Проверить выполнение условий:

а) если $f(y_j) < f(x_k)$, шаг удачный. Положить $x_{k+1} = y_j$, $t_{k+l} = t_k$, $k = k+1$ и проверить условие окончания. Если $k < N$, положить $j = 1$ и перейти к шагу 2. Если $k = N$, поиск завершить: $x^* \cong x_k$, $f(y_j) \cong f(x^\kappa)$;

б) если $f(y^i) > f(x^\kappa)$, шаг неудачный, и перейти к шагу 5.

Шаг 5. Оценить число неудачных шагов из текущей точки:

а) если $j < M$, следует положить $j = j+1$ и перейти к шагу 2;

б) если $j = M$, проверить условие окончания:

- если $t_k \le R_9$ процесс закончить: $x^* \cong x_k$, $f(y_j) \cong f(x^\kappa)$;

- если $t_k > R$, положить $t_k = \beta t_k$, $j = 1$ и перейти к шагу 2.

1.2 Ограничение несанкционированного доступа к тексту программного кода

1.2.1 Скрыть фрагмент кода и ограничить доступ к нему в программном модуле в пакете MathCAD 15

Инструменты пакета MathCAD 15 позволяет скрыть и защитить от несанкционированного доступа информацию, содержащуюся в выбранном фрагменте кода программного модуля.

Последовательность действий:

1. Загрузить шаблон границ области (Рис. 1.3) (путь в главном меню: меню **Вставка** ► строка **Область** (Рис. 1.2)).

2. Заключить требуемый фрагмент кода между верхней и нижней границами. Для этого необходимо выделить требуемую граница, щелкнув на ней левой кнопкой мыши и перетащить ее в необходимое место (Рис. 1.4).

3. Скрыть фрагмент кода между границами. Для этого дважды щелкнуть левой кнопкой мыши на любой из границ (Рис. 1.5).

4. Ограничить несанкционированный доступ к скрытому фрагменту кода (Рис. 1.8). Для этого:

- загрузить диалоговое окно **Блокировка области** (Рис. 1.7) (путь: навести курсор на границы скрытого текста ► нажать правую кнопку мыши ► из разрывающегося списка выбрать строку **Блокировать** (Рис. 1.6)), в котором;
- ввести кодовое слово в строке **Пароль** и подтвердить его;
- снять флаг **Разрешить развертывание и свертывание области во время блокировки**;
- включать флаг **Показать время блокировка**.

Для разблокировки области необходимо:

- навести курсор на границы скрытого текста;
- нажать правую кнопку мыши;
- из раскрывающегося списка выбрать строку **Разблокировать**;
- загрузить диалоговое окно **Снятие блокировки области** (Рис. 1.9);
- ввести кодовое слово в строке **Пароль**.

Удаление границ области блокировки:

- выделить границу области блокировки;
- удалить одну из границ.

Если область фрагмента кода скрыта, необходимо раскрыть область блокировки, а затем ее удалить. В противном случае границы будут удалены вместе с текстом кода.

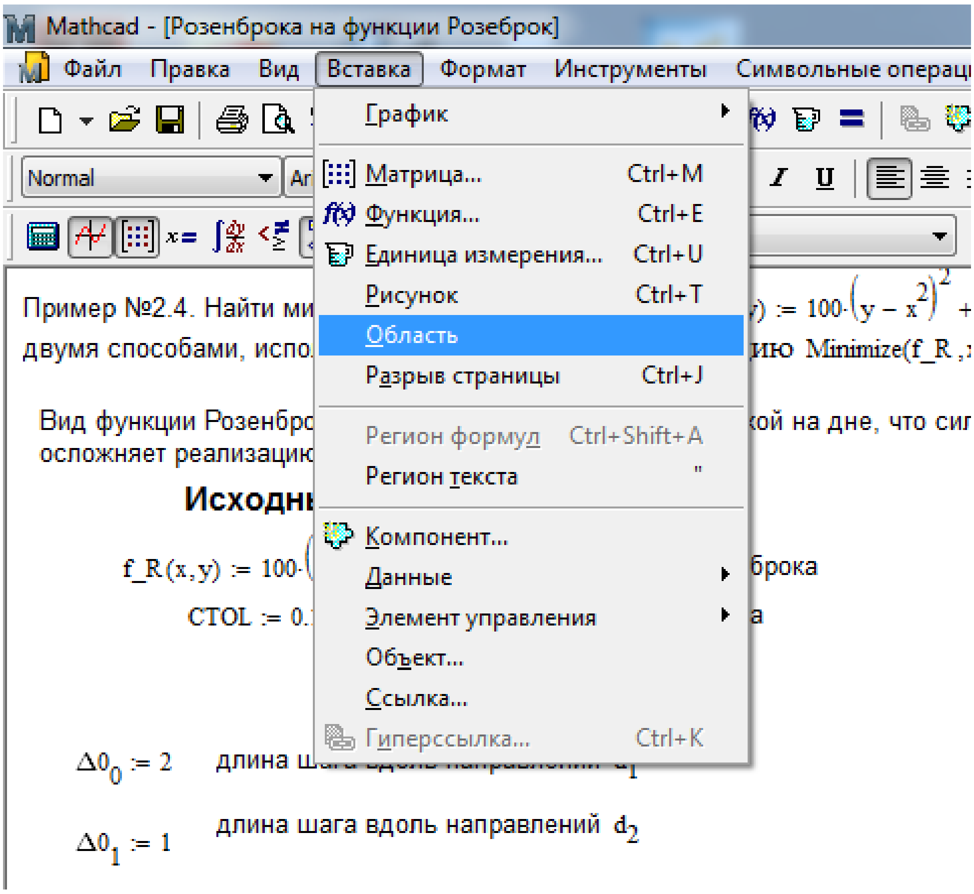


Рис. 1.2. Доступ к команде загрузки шаблона границ области

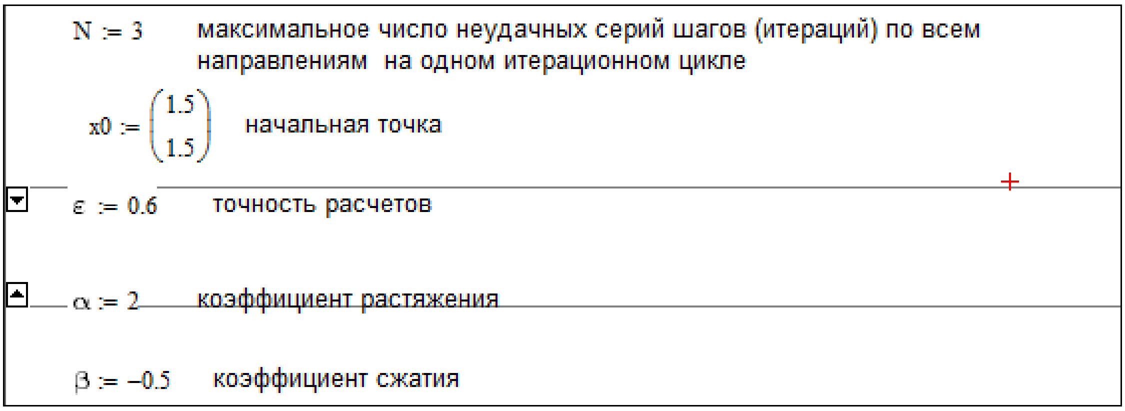


Рис. 1.3. Вид шаблона границ области

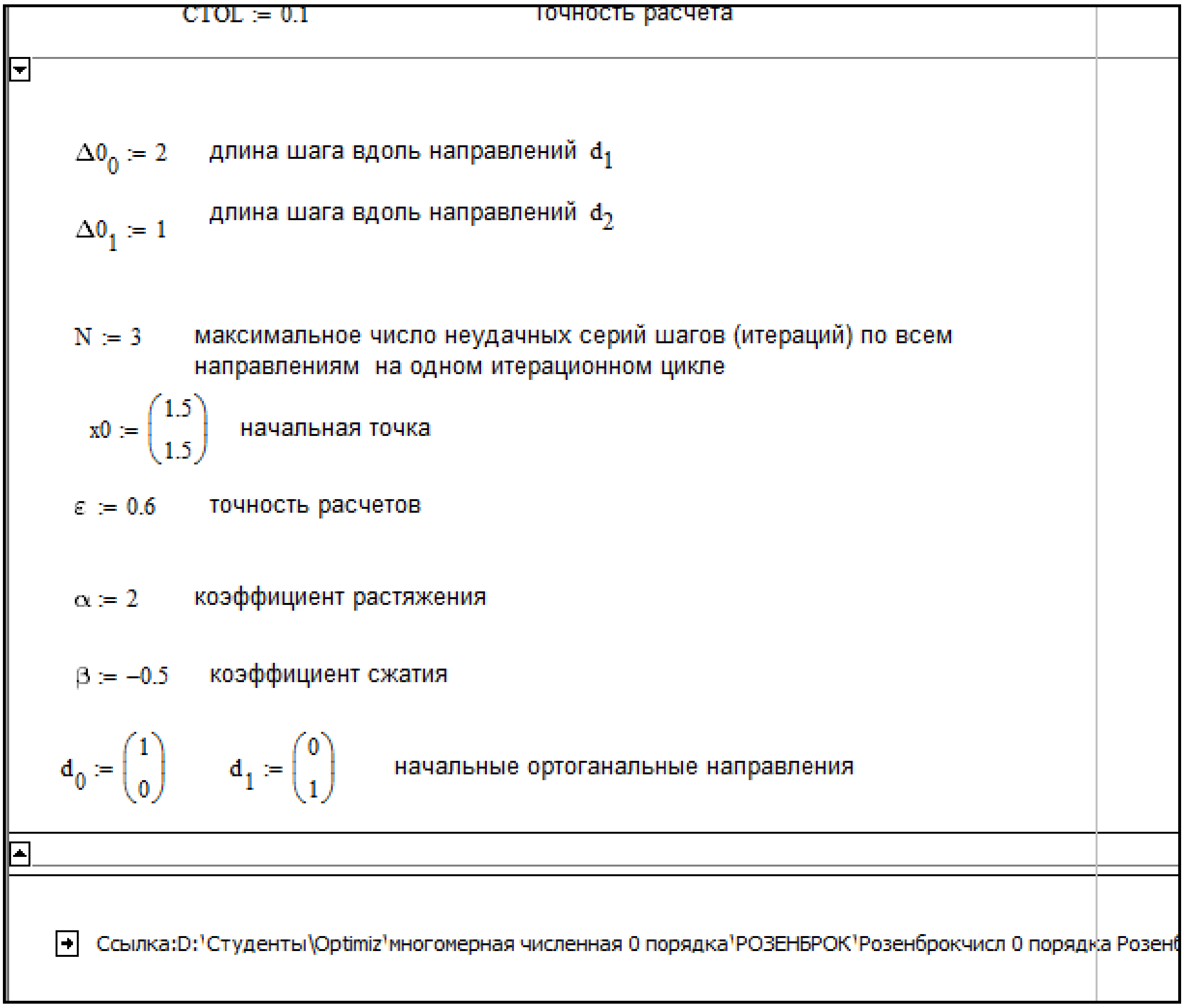


Рис. 1.4. Область фрагмента кода между границами

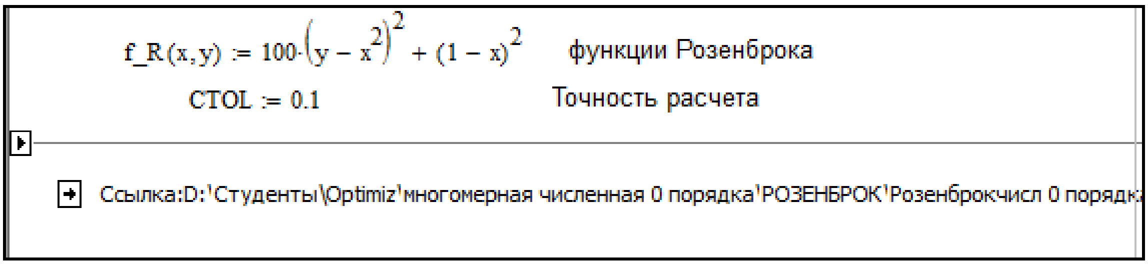


Рис. 1.5. Вид документа со скрытым фрагментом кода

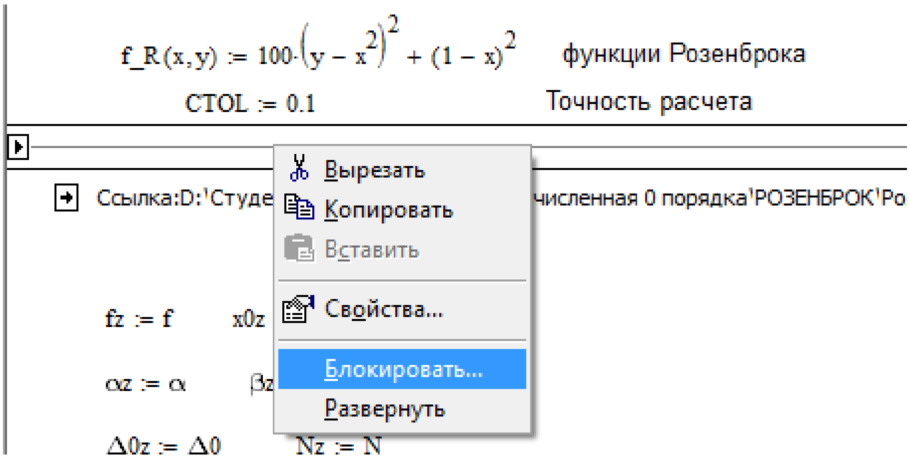


Рис. 1.6. Доступ команде **Блокировать** для ограничения просмотра скрытого фрагмента кода

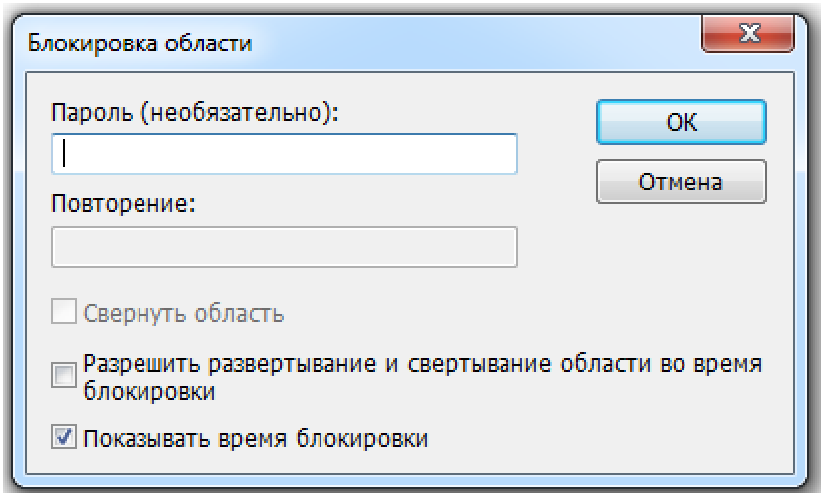


Рис. 1.7. Диалоговое окно **Блокировка** области

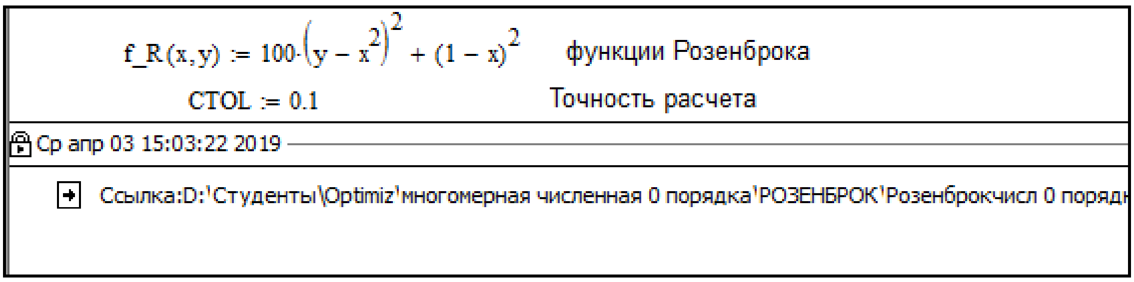


Рис. 1.8. – Вид документа с ограниченным доступом к скрытому фрагменту кода

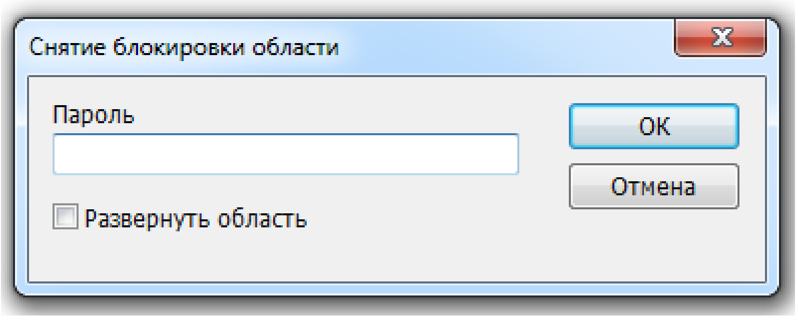


Рис. 1.9. Диалоговое окно **Снятие** блокировки области

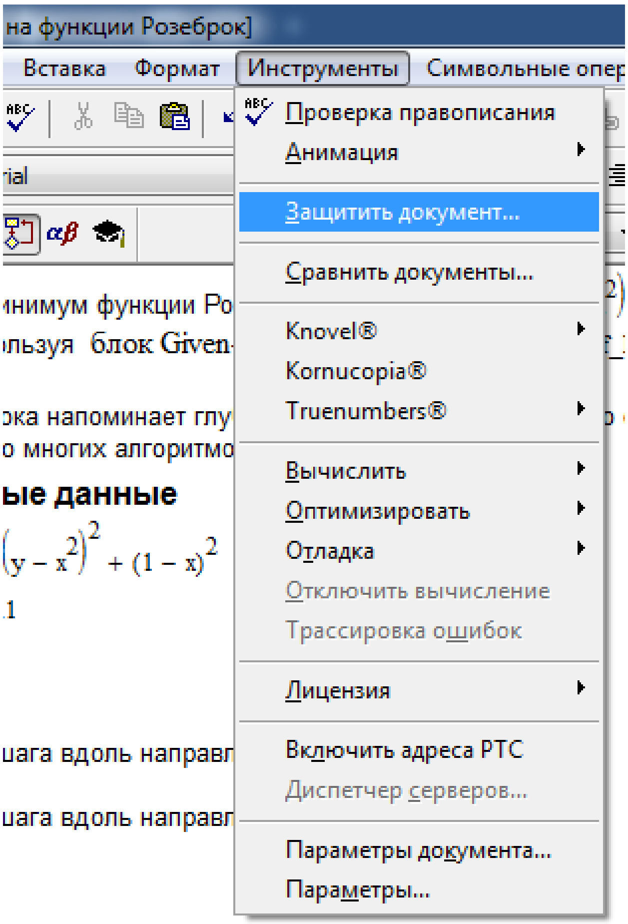


Рис. 1.10. Доступ команде **Защита документа** для ограничения редактирования программного кода

1.2.2 Ограничение редактирования текста программного кода

Инструменты пакета MathCAD 15 позволяет ограничить несанкционированное редактирование кода программного модуля.

Последовательность действий:

- загрузить диалоговое окно Защита документа (Рис. 1.11). (Путь: Главное меню ► вкладка **Инструменты** ► строка **Защитить документ**);

- ввести кодовое слово в строке **Пароль**.

Для разблокирования ограничения на редактирование необходимо:

- загрузить диалоговое окно **Снятие защиты документа** (Путь: Главное меню ► вкладка **Инструменты** ► строка **Снять защиту документа**);

- ввести кодовое слово.

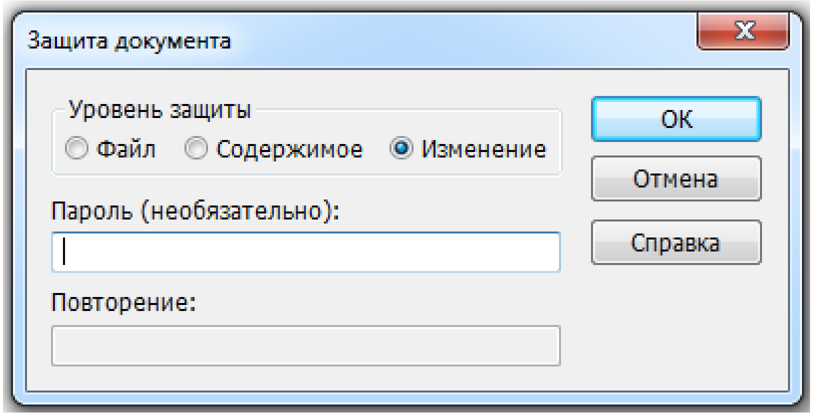


Рис. 1.11. Диалоговое окно **Защита документа**

1.3 Примеры реализации алгоритма метода случайного поиска с возвратом при неудачном шаге

Пример 1.1. Найти минимум функции $f(x) = 4(x_0 - 5)^2 + (x_1 - 6)^2$ методом случайного поиска с возвратом при неудачном шаге

1. Исходные данные и обозначения переменных: x_0 – начальная точка, β – коэффициент сжатия $(0 < \beta < 1)$, M – максимальное число неудачно выполненных испытаний при расчете случайной расчетной точки y_j на текущей итерации по x, $t0$ – начальная величина шага, R – минимальная величину шага, N – максимальное число итераций по x, j – индекс итерации случайной расчетной точки y_j, k – индекс итерации точек x_k при удачном шаге, y – вектор координат случайных точек расчета.

Положить: $x_0 = \begin{pmatrix} 8 \\ 9 \end{pmatrix}$, $\beta = 0.5$, $M = 4$, $R = 0.5$, $t_0 = 1$, $k = 0$, $j = 1$.

2^0. Получить случайный вектор $\varsigma_j = (\varsigma_{j_1}\ \varsigma_{j_2})^T$, где ς_{j_i} – случайная величина, равномерно распределенная на интервале $[-1, 1]$, $n = 2$ – количество элементов в векторе. $\varsigma_j = \begin{pmatrix} 0.648 \\ -0.652 \end{pmatrix}$. Вычислить случайный вектор единичной длины $\xi_j = \dfrac{\varsigma_j}{|\varsigma_j|} = \begin{pmatrix} 0.704 \\ -0.71 \end{pmatrix}$, $\|\xi_j\| = 1$.

3^0. Вычислить координаты случайного вектора $y_j = x_k + t_k \xi_j = \begin{pmatrix} 8.704 \\ 8.29 \end{pmatrix}$.

4^0. Так как $f(y_j) = 60.116 > f(x_k) = 45$, шаг не удачен.

5^0. Оценить число неудачных попыток. Так как $j = 1 < M = 4$, положить $j = j + 1 = 2$. Перейдём к шагу 2.

2^1. Получить случайный вектор $\varsigma_j = (\varsigma_{j_1}\ \varsigma_{j_2})^T$, где ς_{j_i} – случайная величина, равномерно распределенная на интервале $[-1, 1]$, $n = 2$ – количество элементов в векторе. $\varsigma_j = \begin{pmatrix} 0.315 \\ 0.684 \end{pmatrix}$. Вычислить случайный вектор единичной длины $\xi_j = \dfrac{\varsigma_j}{|\varsigma_j|} = \begin{pmatrix} 0.418 \\ 0.908 \end{pmatrix}$, $\|\xi_j\| = 1$.

3^1. Вычислить координаты случайного вектора $y_j = x_k + t_k \xi_j = \begin{pmatrix} 8.418 \\ 9.908 \end{pmatrix}$.

4^1. Так как $f(y_j) = 62.014 > f(x_k) = 45$, шаг не удачен.

5^1. Оценить число неудачных попыток. Так как $j = 2 < M = 4$, положить $j = j + 1 = 3$. Перейдём к шагу 2.

2^2. Получить случайный вектор $\varsigma_j = (\varsigma_{j_1}\ \varsigma_{j_2})^T$, где ς_{j_i} – случайная величина, равномерно распределенная на интервале $[-1, 1]$, $n = 2$ – количество элементов в векторе. $\varsigma_j = \begin{pmatrix} -0.817 \\ -0.705 \end{pmatrix}$. Вычислить случайный вектор единичной длины $\xi_j = \dfrac{\varsigma_j}{|\varsigma_j|} = \begin{pmatrix} -0.757 \\ -0.653 \end{pmatrix}$, $\|\xi_j\| = 1$.

3^2. Вычислить координаты случайного вектора $y_j = x_k + t_k \xi_j = \begin{pmatrix} 7.243 \\ 8.347 \end{pmatrix}$.

4^2. Так как $f(y_j) = 25.629 < f(x_k) = 45$, шаг удачен.

Положить $x_{k+1} = y_j = \begin{pmatrix} 7.243 \\ 8.347 \end{pmatrix}$, $t_{k+1} = t_k = 1$, $k = k+1 = 1$.

Проверить условие окончания расчета по N. Так как $k = 1 < N = 10$, положить $j = 1$ и перейти к шагу 2.

2^3. Получить случайный вектор $\varsigma_j = (\varsigma_{j_1}\ \varsigma_{j_2})^T$, где ς_{j_i} – случайная величина, равномерно распределенная на интервале $[-1, 1]$, $n = 2$ – количество элементов в векторе. $\varsigma_j = \begin{pmatrix} -0.982 \\ 0.063 \end{pmatrix}$. Вычислить случайный вектор единичной длины $\xi_j = \dfrac{\varsigma_j}{|\varsigma_j|} = \begin{pmatrix} -0.998 \\ 0.064 \end{pmatrix}$, $\|\xi_j\| = 1$.

3^3. Вычислить координаты случайного вектора $y_j = x_k + t_k \xi_j = \begin{pmatrix} 6.245 \\ 8.411 \end{pmatrix}$.

4^3. Так как $f(y_j) = 12.011 < f(x_k) = 25.629$, шаг удачен.

Положить $x_{k+1} = y_j = \begin{pmatrix} 6.245 \\ 8.411 \end{pmatrix}$, $t_{k+1} = t_k = 1$, $k = k+1 = 2$.

Проверить условие окончания расчета по N. Так как $k = 2 < N = 10$, положить $j = 1$ и перейти к шагу 2.

2^4. Получить случайный вектор $\varsigma_j = (\varsigma_{j_1}\ \varsigma_{j_2})^T$, где ς_{j_i} – случайная величина, равномерно распределенная на интервале $[-1, 1]$, $n = 2$ – количество элементов в векторе. $\varsigma_j = \begin{pmatrix} -0.098 \\ -0.886 \end{pmatrix}$. Вычислить случайный вектор единичной длины $\xi_j = \dfrac{\varsigma_j}{|\varsigma_j|} = \begin{pmatrix} -0.11 \\ -0.994 \end{pmatrix}$, $\|\xi_j\| = 1$.

3^4. Вычислить координаты случайного вектора $y_j = x_k + t_k \xi_j = \begin{pmatrix} 6.135 \\ 7.417 \end{pmatrix}$.

4^4. Так как $f(y_j) = 7.16 < f(x_k) = 12.011$, шаг удачен.

Положить $x_{k+1} = y_j = \begin{pmatrix} 6.135 \\ 7.417 \end{pmatrix}$, $t_{k+1} = t_k = 1$, $k = k+1 = 3$.

Проверить условие окончания расчета по N. Так как $k = 3 < N = 10$, положить $j = 1$ и перейти к шагу 2.

2^5. Получить случайный вектор $\varsigma_j = (\varsigma_{j_1}\ \varsigma_{j_2})^T$, где ς_{j_i} – случайная величина, равномерно распределенная на интервале $[-1,1]$, $n=2$ – количество элементов в векторе. $\varsigma_j = \begin{pmatrix} 0.752 \\ 0.912 \end{pmatrix}$. Вычислить случайный вектор единичной длины $\xi_j = \dfrac{\varsigma_j}{|\varsigma_j|} = \begin{pmatrix} 0.636 \\ 0.772 \end{pmatrix}$, $\|\xi_j\| = 1$.

3^5. Вычислить координаты случайного вектора $y_j = x_k + t_k \xi_j = \begin{pmatrix} 6.771 \\ 8.188 \end{pmatrix}$.

4^5. Так как $f(y_j) = 17.337 > f(x_k) = 7.16$, шаг не удачен.

5^2. Оценить число неудачных попыток. Так как $j = 1 < M = 4$, положить $j = j + 1 = 2$. Перейдём к шагу 2.

2^6. Получить случайный вектор $\varsigma_j = (\varsigma_{j_1}\ \varsigma_{j_2})^T$, где ς_{j_i} – случайная величина, равномерно распределенная на интервале $[-1,1]$, $n=2$ – количество элементов в векторе. $\varsigma_j = \begin{pmatrix} 0.724 \\ 0.559 \end{pmatrix}$. Вычислить случайный вектор единичной длины $\xi_j = \dfrac{\varsigma_j}{|\varsigma_j|} = \begin{pmatrix} 0.792 \\ 0.611 \end{pmatrix}$, $\|\xi_j\| = 1$.

3^6. Вычислить координаты случайного вектора $y_j = x_k + t_k \xi_j = \begin{pmatrix} 6.927 \\ 8.028 \end{pmatrix}$.

4^6. Так как $f(y_j) = 18.959 > f(x_k) = 7.16$, шаг не удачен.

5^3. Оценить число неудачных попыток. Так как $j = 2 < M = 4$, положить $j = j + 1 = 3$. Перейдём к шагу 2.

2^7. Получить случайный вектор $\varsigma_j = (\varsigma_{j_1}\ \varsigma_{j_2})^T$, где ς_{j_i} – случайная величина, равномерно распределенная на интервале $[-1,1]$, $n=2$ – количество элементов в векторе. $\varsigma_j = \begin{pmatrix} 0.17 \\ -0.299 \end{pmatrix}$. Вычислить случайный вектор единичной длины $\xi_j = \dfrac{\varsigma_j}{|\varsigma_j|} = \begin{pmatrix} 0.494 \\ -0.869 \end{pmatrix}$, $\|\xi_j\| = 1$.

3^7. Вычислить координаты случайного вектора $y_j = x_k + t_k \xi_j = \begin{pmatrix} 6.629 \\ 6.547 \end{pmatrix}$.

4^7. Так как $f(y_j) = 10.918 > f(x_k) = 7.16$, шаг не удачен.

5^4. Оценить число неудачных попыток. Так как $j = 3 < M = 4$, положить $j = j + 1 = 4$. Перейдём к шагу 2.

2^8. Получить случайный вектор $\varsigma_j = (\varsigma_{j_1}\ \varsigma_{j_2})^T$, где ς_{j_i} – случайная величина, равномерно распределенная на интервале $[-1, 1]$, $n = 2$ – количество элементов в векторе. $\varsigma_j = \begin{pmatrix} -0.817 \\ -0.705 \end{pmatrix}$. Вычислить случайный вектор единичной длины $\xi_j = \dfrac{\varsigma_j}{|\varsigma_j|} = \begin{pmatrix} -0.757 \\ -0.653 \end{pmatrix}$, $\|\xi_j\| = 1$.

3^8. Вычислить координаты случайного вектора $y_j = x_k + t_k \xi_j = \begin{pmatrix} 5.378 \\ 6.763 \end{pmatrix}$.

4^8. Так как $f(y_j) = 1.154 < f(x_k) = 7.16$, шаг удачен.

Положить $x_{k+1} = y_j = \begin{pmatrix} 5.378 \\ 6.763 \end{pmatrix}$, $t_{k+1} = t_k = 1$, $k = k + 1 = 4$.

Проверить условие окончания расчета по N. Так как $k = 4 < N = 10$, положить $j = 1$ и перейти к шагу 2.

2^9. Получить случайный вектор $\varsigma_j = (\varsigma_{j_1}\ \varsigma_{j_2})^T$, где ς_{j_i} – случайная величина, равномерно распределенная на интервале $[-1, 1]$, $n = 2$ – количество элементов в векторе. $\varsigma_j = \begin{pmatrix} -0.982 \\ 0.063 \end{pmatrix}$. Вычислить случайный вектор единичной длины $\xi_j = \dfrac{\varsigma_j}{|\varsigma_j|} = \begin{pmatrix} -0.998 \\ 0.064 \end{pmatrix}$, $\|\xi_j\| = 1$.

3^9. Вычислить координаты случайного вектора $y_j = x_k + t_k \xi_j = \begin{pmatrix} 4.38 \\ 6.827 \end{pmatrix}$.

4^9. Так как $f(y_j) = 2.222 > f(x_k) = 1.154$, шаг не удачен.

5^5. Оценить число неудачных попыток. Так как $j = 1 < M = 4$, положить $j = j + 1 = 2$. Перейдём к шагу 2.

2^{10}. Получить случайный вектор $\varsigma_j = (\varsigma_{j_1}\ \varsigma_{j_2})^T$, где ς_{j_i} – случайная величина, равномерно распределенная на интервале $[-1, 1]$, $n = 2$ – количество

элементов в векторе. $\varsigma_j = \begin{pmatrix} -0.098 \\ -0.886 \end{pmatrix}$. Вычислить случайный вектор единичной длины $\xi_j = \dfrac{\varsigma_j}{|\varsigma_j|} = \begin{pmatrix} -0.11 \\ -0.994 \end{pmatrix}$, $\|\xi_j\| = 1$.

3^{10}. Вычислить координаты случайного вектора $y_j = x_k + t_k \xi_j = \begin{pmatrix} 5.268 \\ 5.77 \end{pmatrix}$.

4^{10}. Так как $f(y_j) = 0.34 > f(x_k) = 1.154$, шаг удачен.

Положить $x_{k+1} = y_j = \begin{pmatrix} 5.268 \\ 5.77 \end{pmatrix}$, $t_{k+1} = t_k = 1$, $k = k+1 = 5$.

Проверить условие окончания расчета по N. Так как $k = 5 < N = 10$, положить $j = 1$ и перейти к шагу 2.

2^{11}. Получить случайный вектор $\varsigma_j = (\varsigma_{j_1}\ \varsigma_{j_2})^T$, где ς_{j_i} – случайная величина, равномерно распределенная на интервале $[-1, 1]$, $n = 2$ – количество элементов в векторе. $\varsigma_j = \begin{pmatrix} 0.752 \\ 0.912 \end{pmatrix}$. Вычислить случайный вектор единичной длины $\xi_j = \dfrac{\varsigma_j}{|\varsigma_j|} = \begin{pmatrix} 0.636 \\ 0.772 \end{pmatrix}$, $\|\xi_j\| = 1$.

3^{11}. Вычислить координаты случайного вектора $y_j = x_k + t_k \xi_j = \begin{pmatrix} 4.38 \\ 6.827 \end{pmatrix}$.

4^{11}. Так как $f(y_j) = 3.563 > f(x_k) = 0.34$, шаг не удачен.

5^{6}. Оценить число неудачных попыток. Так как $j = 1 < M = 4$, положить $j = j + 1 = 2$. Перейдём к шагу 2.

2^{12}. Получить случайный вектор $\varsigma_j = (\varsigma_{j_1}\ \varsigma_{j_2})^T$, где ς_{j_i} – случайная величина, равномерно распределенная на интервале $[-1, 1]$, $n = 2$ – количество элементов в векторе. $\varsigma_j = \begin{pmatrix} 0.994 \\ 0.223 \end{pmatrix}$. Вычислить случайный вектор единичной длины $\xi_j = \dfrac{\varsigma_j}{|\varsigma_j|} = \begin{pmatrix} 0.976 \\ 0.219 \end{pmatrix}$, $\|\xi_j\| = 1$.

3^{12}. Вычислить координаты случайного вектора $y_j = x_k + t_k \xi_j = \begin{pmatrix} 6.244 \\ 5.988 \end{pmatrix}$.

4^{12}. Так как $f(y_j) = 6.188 > f(x_k) = 0.34$, шаг не удачен.

5^7. Оценить число неудачных попыток. Так как $j = 2 < M = 4$, положить $j = j + 1 = 3$. Перейдём к шагу 2.

2^{13}. Получить случайный вектор $\varsigma_j = (\varsigma_{j_1}\ \varsigma_{j_2})^T$, где ς_{j_i} – случайная величина, равномерно распределенная на интервале $[-1, 1]$, $n = 2$ – количество элементов в векторе. $\varsigma_j = \begin{pmatrix} -0.248 \\ 0.354 \end{pmatrix}$. Вычислить случайный вектор единичной длины $\xi_j = \dfrac{\varsigma_j}{|\varsigma_j|} = \begin{pmatrix} -0.574 \\ 0.819 \end{pmatrix}$, $\|\xi_j\| = 1$.

3^{13}. Вычислить координаты случайного вектора $y_j = x_k + t_k \xi_j = \begin{pmatrix} 4.694 \\ 6.589 \end{pmatrix}$.

4^{13}. Так как $f(y_j) = 0.72 > f(x_k) = 0.34$, шаг не удачен.

5^8. Оценить число неудачных попыток. Так как $j = 3 < M = 4$, положить $j = j + 1 = 4$. Перейдём к шагу 2.

2^{14}. Получить случайный вектор $\varsigma_j = (\varsigma_{j_1}\ \varsigma_{j_2})^T$, где ς_{j_i} – случайная величина, равномерно распределенная на интервале $[-1, 1]$, $n = 2$ – количество элементов в векторе. $\varsigma_j = \begin{pmatrix} 0.176 \\ 0.675 \end{pmatrix}$. Вычислить случайный вектор единичной длины $\xi_j = \dfrac{\varsigma_j}{|\varsigma_j|} = \begin{pmatrix} 0.252 \\ 0.968 \end{pmatrix}$, $\|\xi_j\| = 1$.

3^{14}. Вычислить координаты случайного вектора $y_j = x_k + t_k \xi_j = \begin{pmatrix} 5.52 \\ 6.737 \end{pmatrix}$.

4^{14}. Так как $f(y_j) = 1.626 > f(x_k) = 0.34$, шаг не удачен.

5^9. Оценить число неудачных попыток. Так как $j = 4 = M = 4$, проверить условия окончания расчета по R. Так как $t_k \le R$, положить $t_k = \beta\, t_k = 0.5$ и $j = 1$

2^{15}. Получить случайный вектор $\varsigma_j = (\varsigma_{j_1}\ \varsigma_{j_2})^T$, где ς_{j_i} – случайная величина, равномерно распределенная на интервале $[-1, 1]$, $n = 2$ – количество элементов в векторе. $\varsigma_j = \begin{pmatrix} -0.705 \\ -0.717 \end{pmatrix}$. Вычислить случайный вектор единичной длины $\xi_j = \dfrac{\varsigma_j}{|\varsigma_j|} = \begin{pmatrix} -0.701 \\ -0.713 \end{pmatrix}$, $\|\xi_j\| = 1$.

3^{15}. Вычислить координаты случайного вектора $y_j = x_k + t_k\xi_j = \begin{pmatrix} 4.917 \\ 5.413 \end{pmatrix}$.

4^{15}. Так как $f(y_j) = 0.372 > f(x_k) = 0.34$, шаг не удачен.

5^{10}. Оценить число неудачных попыток. Так как $j = 1 < M = 4$, положить $j = j + 1 = 2$. Перейдём к шагу 2.

2^{16}. Получить случайный вектор $\varsigma_j = (\varsigma_{j_1}\ \varsigma_{j_2})^T$, где ς_{j_i} – случайная величина, равномерно распределенная на интервале $[-1, 1]$, $n = 2$ – количество элементов в векторе. $\varsigma_j = \begin{pmatrix} 0.933 \\ -0.693 \end{pmatrix}$. Вычислить случайный вектор единичной длины $\xi_j = \dfrac{\varsigma_j}{|\varsigma_j|} = \begin{pmatrix} 0.803 \\ -0.596 \end{pmatrix}$, $\|\xi_j\| = 1$.

3^{16}. Вычислить координаты случайного вектора $y_j = x_k + t_k\xi_j = \begin{pmatrix} 5.669 \\ 5.471 \end{pmatrix}$.

4^{16}. Так как $f(y_j) = 2.072 > f(x_k) = 0.34$, шаг не удачен.

5^{11}. Оценить число неудачных попыток. Так как $j = 2 < M = 4$, положить $j = j + 1 = 3$. Перейдём к шагу 2.

2^{17}. Получить случайный вектор $\varsigma_j = (\varsigma_{j_1}\ \varsigma_{j_2})^T$, где ς_{j_i} – случайная величина, равномерно распределенная на интервале $[-1, 1]$, $n = 2$ – количество элементов в векторе. $\varsigma_j = \begin{pmatrix} 0.464 \\ -0.441 \end{pmatrix}$. Вычислить случайный вектор единичной длины $\xi_j = \dfrac{\varsigma_j}{|\varsigma_j|} = \begin{pmatrix} 0.725 \\ -0.689 \end{pmatrix}$, $\|\xi_j\| = 1$.

3^{17}. Вычислить координаты случайного вектора $y_j = x_k + t_k\xi_j = \begin{pmatrix} 5.63 \\ 5.425 \end{pmatrix}$.

4^{17}. Так как $f(y_j) = 1.92 > f(x_k) = 0.34$, шаг не удачен.

5^{12}. Оценить число неудачных попыток. Так как $j = 3 < M = 4$, положить $j = j + 1 = 4$. Перейдём к шагу 2.

2^{18}. Получить случайный вектор $\varsigma_j = (\varsigma_{j_1}\ \varsigma_{j_2})^T$, где ς_{j_i} – случайная величина, равномерно распределенная на интервале $[-1, 1]$, $n = 2$ – количество

элементов в векторе. $\varsigma_j = \begin{pmatrix} -0.084 \\ 0.489 \end{pmatrix}$. Вычислить случайный вектор единичной длины $\xi_j = \dfrac{\varsigma_j}{|\varsigma_j|} = \begin{pmatrix} -0.169 \\ 0.986 \end{pmatrix}$, $\|\xi_j\| = 1$.

3^{18}. Вычислить координаты случайного вектора $y_j = x_k + t_k \xi_j = \begin{pmatrix} 5.183 \\ 6.262 \end{pmatrix}$.

4^{18}. Так как $f(y_j) = 0.203 < f(x_k) = 0.34$, шаг удачен.

Положить $x_{k+1} = y_j = \begin{pmatrix} 5.183 \\ 6.262 \end{pmatrix}$, $t_{k+1} = t_k = 0.5$, $k = k+1 = 6$.

Проверить условие окончания расчета по N. Так как $k = 6 < N = 10$, положить $j = 1$ и перейти к шагу 2.

2^{19}. Получить случайный вектор $\varsigma_j = (\varsigma_{j_1}\ \varsigma_{j_2})^T$, где ς_{j_i} – случайная величина, равномерно распределенная на интервале $[-1, 1]$, $n = 2$ – количество элементов в векторе. $\varsigma_j = \begin{pmatrix} 0.145 \\ -0.697 \end{pmatrix}$. Вычислить случайный вектор единичной длины $\xi_j = \dfrac{\varsigma_j}{|\varsigma_j|} = \begin{pmatrix} 0.204 \\ -0.979 \end{pmatrix}$, $\|\xi_j\| = 1$.

3^{19}. Вычислить координаты случайного вектора $y_j = x_k + t_k \xi_j = \begin{pmatrix} 5.285 \\ 5.773 \end{pmatrix}$.

4^{19}. Так как $f(y_j) = 0.377 > f(x_k) = 0.203$, шаг не удачен.

5^{13}. Оценить число неудачных попыток. Так как $j = 1 < M = 4$, положить $j = j + 1 = 2$. Перейдём к шагу 2.

2^{20}. Получить случайный вектор $\varsigma_j = (\varsigma_{j_1}\ \varsigma_{j_2})^T$, где ς_{j_i} – случайная величина, равномерно распределенная на интервале $[-1, 1]$, $n = 2$ – количество элементов в векторе. $\varsigma_j = \begin{pmatrix} -0.705 \\ -0.717 \end{pmatrix}$. Вычислить случайный вектор единичной длины $\xi_j = \dfrac{\varsigma_j}{|\varsigma_j|} = \begin{pmatrix} -0.701 \\ -0.713 \end{pmatrix}$, $\|\xi_j\| = 1$.

3^{20}. Вычислить координаты случайного вектора $y_j = x_k + t_k \xi_j = \begin{pmatrix} 4.833 \\ 5.906 \end{pmatrix}$.

4^{20}. Так как $f(y_j) = 0.121 < f(x_k) = 0.203$, шаг удачен.

Положить $x_{k+1} = y_j = \begin{pmatrix} 4.833 \\ 5.906 \end{pmatrix}$, $t_{k+1} = t_k = 0.5$, $k = k+1 = 7$.

Проверить условие окончания расчета по N. Так как $k = 7 < N = 10$, положить $j = 1$ и перейти к шагу 2.

2^{21}. Получить случайный вектор $\varsigma_j = (\varsigma_{j_1}\ \varsigma_{j_2})^T$, где ς_{j_i} – случайная величина, равномерно распределенная на интервале $[-1, 1]$, $n = 2$ – количество элементов в векторе. $\varsigma_j = \begin{pmatrix} 0.364 \\ 0.444 \end{pmatrix}$. Вычислить случайный вектор единичной длины $\xi_j = \dfrac{\varsigma_j}{|\varsigma_j|} = \begin{pmatrix} 0.634 \\ 0.773 \end{pmatrix}$, $\|\xi_j\| = 1$.

3^{21}. Вычислить координаты случайного вектора $y_j = x_k + t_k \xi_j = \begin{pmatrix} 5.15 \\ 6.292 \end{pmatrix}$.

4^{21}. Так как $f(y_j) = 0.175 > f(x_k) = 0.121$, шаг не удачен.

5^{14}. Оценить число неудачных попыток. Так как $j = 1 < M = 4$, положить $j = j + 1 = 2$. Перейдём к шагу 2.

2^{22}. Получить случайный вектор $\varsigma_j = (\varsigma_{j_1}\ \varsigma_{j_2})^T$, где ς_{j_i} – случайная величина, равномерно распределенная на интервале $[-1, 1]$, $n = 2$ – количество элементов в векторе. $\varsigma_j = \begin{pmatrix} -0.754 \\ 0.669 \end{pmatrix}$. Вычислить случайный вектор единичной длины $\xi_j = \dfrac{\varsigma_j}{|\varsigma_j|} = \begin{pmatrix} -0.748 \\ 0.664 \end{pmatrix}$, $\|\xi_j\| = 1$.

3^{22}. Вычислить координаты случайного вектора $y_j = x_k + t_k \xi_j = \begin{pmatrix} 4.459 \\ 6.238 \end{pmatrix}$.

4^{22}. Так как $f(y_j) = 1.228 > f(x_k) = 0.121$, шаг не удачен.

5^{15}. Оценить число неудачных попыток. Так как $j = 2 < M = 4$, положить $j = j + 1 = 3$. Перейдём к шагу 2.

2^{23}. Получить случайный вектор $\varsigma_j = (\varsigma_{j_1}\ \varsigma_{j_2})^T$, где ς_{j_i} – случайная величина, равномерно распределенная на интервале $[-1, 1]$, $n = 2$ – количество

элементов в векторе. $\varsigma_j = \begin{pmatrix} 0.034 \\ -0.148 \end{pmatrix}$. Вычислить случайный вектор единичной длины $\xi_j = \frac{\varsigma_j}{|\varsigma_j|} = \begin{pmatrix} 0.224 \\ -0.975 \end{pmatrix}$, $\|\xi_j\| = 1$.

3^{23}. Вычислить координаты случайного вектора $y_j = x_k + t_k \xi_j = \begin{pmatrix} 4.945 \\ 5.418 \end{pmatrix}$.

4^{23}. Так как $f(y_j) = 0.35 > f(x_k) = 0.121$, шаг не удачен.

5^{16}. Оценить число неудачных попыток. Так как $j = 3 < M = 4$, положить $j = j + 1 = 4$. Перейдём к шагу 2.

2^{24}. Получить случайный вектор $\varsigma_j = (\varsigma_{j_1}\ \varsigma_{j_2})^T$, где ς_{j_i} – случайная величина, равномерно распределенная на интервале $[-1, 1]$, $n = 2$ – количество элементов в векторе. $\varsigma_j = \begin{pmatrix} 0.899 \\ 0.099 \end{pmatrix}$. Вычислить случайный вектор единичной длины $\xi_j = \frac{\varsigma_j}{|\varsigma_j|} = \begin{pmatrix} 0.994 \\ 0.109 \end{pmatrix}$, $\|\xi_j\| = 1$.

3^{24}. Вычислить координаты случайного вектора $y_j = x_k + t_k \xi_j = \begin{pmatrix} 5.33 \\ 5.961 \end{pmatrix}$.

4^{24}. Так как $f(y_j) = 0.437 > f(x_k) = 0.121$, шаг не удачен.

5^{16}. Оценить число неудачных попыток. Так как $j = 4 = M = 4$, проверить условия окончания расчета по R. Так как $t_k = R$ расчет завершен. $x^* \cong x_k = \begin{pmatrix} 4.833 \\ 5.906 \end{pmatrix}$ – координаты точки минимума, $f\left(x^*\right) = 0.121$ – значение функции в точке минимума.

Листинг с программой расчета и графиками выполнен в MathCAD 15 и приведен на Рис. 2.1–Рис. 2.40 на стр. 38–77.

Листинг с описанием собственных функций, реализующих метод случайного поиска с возвратом при неудачном шаге в MathCAD 15, приведен на Рис. 2.41–Рис. 2.43 на стр. 78–80.

Следует отметить, что при описании алгоритма расчета и его реализации для функции в примерах индекс первого элемента используемых векторов равен 1, а при реализации в пакете MathCAD 15 индекс первого элемента векторов равен 0. Это необходимо помнить при изучении материала.

Необходимо обратить внимание на то, как структурирована программа при построчной реализации метода. Набирать текст программы необходимо в полном объеме только для первой итерации. В дальнейшем надо копировать ранее набранные подходящие фрагменты кода и лишь немного их корректировать.

Внимание: В учебных целях значение для случайного вектора рассчитывается с помощью функции $\mathrm{runif}(2,-1,1)$ на каждом шаге, а затем копируется в переменную ς, и значение для ς становится постоянным для данного примера. В противном случае не удается добиться стабильно работающего алгоритма при построчной реализации для этого метода. При написании пользовательской функции, реализующий алгоритм, этого допущения нет.

Для просмотра результатов расчета при различных вариантах расчетных данных (т.к. используется датчик случайных чисел) необходимо перезапустить пользовательскую функцию $\mathrm{F\_СП\_ВНШ}(fz, x0z, Mz, Nz, t0z, Rz, \beta z, вx)$ (для перезапуска функции необходимо щелкнуть левой кнопкой мыши на имени функции и нажать клавишу F9).

Для уяснения работы алгоритмов, реализующих методы случайного поиска, необходимо несколько раз перезапустить расчет (для этого необходимо нажать комбинацию клавиш «Ctrl+F9».) и посмотреть, как изменится кривая расчета на графике. Проделать указанное для следующих вариантов начальных условий:

x_0	β	$M\varsigma$	t_0	R
$\begin{pmatrix} 7 \\ 7 \end{pmatrix}$	0.75	10	0.75	0.25
$\begin{pmatrix} 10 \\ 9 \end{pmatrix}$	0.25	10	0.5	0.15

Следует отметить, что алгоритм построчной реализации не изменится по причинам, приведенным выше. Для того, чтобы скрыть ненужный фрагмент текста программы, необходимо воспользоваться алгоритмом действий, приведенным в параграфе «Скрыть фрагмент кода и ограничить доступ к нему в программном модуле в пакете MathCAD 15» на стр. 8.

При выполнении примера студент должен знать следующие разделы MathCad 15:

- Ввод переменных, переменных с нижним индексом, сопроводительного текста.
- Размещение переменных в сопроводительном тексте.

- Просмотр результатов расчета и редактирование числа выводимых десятичных знаков.
- Определение однострочной и многострочной функции и ее вызов.
- Векторизация.
- Доступ к элементам массива, в том числе блочного массива.
- Дискретный аргумент.
- Построение и редактирование трехмерных графиков. Нанесение на один график нескольких кривых.
- Загрузка файла с собственными функциями в текст программы.
- Численное решение уравнений (блок **Given–MinErr**).
- Матричные операции и функции.
- Оператор **if(условие, значение, значение)**.
- Включение и выключение вычисления функций и операторов.

Пример 1.2. Найти минимум функции Хаммельблау $f(x)=\left[x_0^2+x_1+11\right]^2+\left[x_0+x_1^2-7\right]^2$ методом случайного поиска с возвратом при неудачном шаге

1. Исходные данные и обозначения переменных: x_0 – начальная точка, β – коэффициент сжатия $(0<\beta<1)$, M – максимальное число неудачно выполненных испытаний, при расчете случайной расчетной точки y_j, на текущей итерации по x, $t0$ – начальная величина шага, R – минимальная величину шага, N – максимальное число итераций по x, j – индекс итерации случайной расчетной точки y_j, k – индекс итерации точек x_k при удачном шаге, y – вектор координат случайных точек расчета.

Положить: $x_0=\begin{pmatrix}0\\-1\end{pmatrix}$, $\beta=0.5$, $M=3$, $R=0.5$, $N=10$, $t_0=1$, $k=0$, $j=1$.

Самостоятельно провести расчет построчной реализации алгоритма метода **случайного поиска с возвратом при неудачном шаге** по аналогии с последовательностью действий, приведенной в примере 1.1.

Листинг с программой расчета и графиками выполнен в MathCAD 15 и приведен на Рис. 2.44–Рис. 2.55 на стр. 81–92.

Листинг с описанием собственных функций, реализующих метод **случайного поиска с возвратом при неудачном шаге** в MathCAD 15 приведен на Рис. 2.41–Рис. 2.43 на стр. 78–80.

Для уяснения работы алгоритмов, реализующих методы случайного поиска в MathCAD 15 надо несколько раз перезапустить расчет (нажать комбинацию клавиш «Ctrl+F9».) и посмотреть, как изменится кривая расчета на графике. Проделать указанное для следующих вариантов начальных условий:

x_0	β	$M\varsigma$	t_0	R
$\begin{pmatrix}0\\-1\end{pmatrix}$	0.5	10	1	0.25
$\begin{pmatrix}-4\\0\end{pmatrix}$	0.5	10	0.75	0.15
$\begin{pmatrix}0\\3\end{pmatrix}$	0.5	10	0.1	0.01

Для того, чтобы скрыть ненужный фрагмент текста программы, расположенный между исходными данными и графиком, необходимо воспользоваться алгоритмом действий, приведенным в параграфе «Скрыть фрагмент кода и ограничить доступ к нему в программном модуле в пакете MathCAD 15» на стр. 8.

Пример 1.3. Найти минимум функции Розенброка $f(x) = 100\left[x_1 - x_0^2\right]^2 + \left(1 - x_0\right)^2$ методом наилучшей пробы

1. Исходные данные и обозначения переменных: x_0 – начальная точка, β – коэффициент сжатия ($0 < \beta < 1$), M – максимальное число испытаний, $t0$ – начальная величина шага, R – минимальная величину шага, N – максимальное число итераций при удачном шаге, j – индекс случайной расчетной точки y_j, k – индекс итерации точек x_k при удачном шаге, y – вектор координат генерируемых случайных точек расчета.

Положить: $x_0 = \begin{pmatrix}2\\2\end{pmatrix}$, $\beta = 0.5$, $M = 3$, $R = 0.0625$, $N = 10$, $t_0 = 1$, $k = 0$, $j = 1$.

<u>**Самостоятельно**</u> провести расчет построчной реализации алгоритма метода **случайного поиска с возвратом при неудачном шаге** по аналогии с последовательностью действий, приведенной в примере 1.1.

Листинг с программой расчета и графиками выполнен в MathCAD 15 и приведен на стр. Рис. 2.56–Рис. 2.65 на стр. 93–102.

Листинг с описанием собственных функций, реализующих метод **случайного поиска с возвратом при неудачном шаге** в MathCAD 15 приведен на Рис. 2.41–Рис. 2.43 на стр. 78–80.

Для уяснения работы алгоритмов, реализующих методы случайного поиска в MathCAD 15 надо несколько раз перезапустить расчет (нажать комбинацию клавиш «Ctrl+F9».) и посмотреть, как изменится кривая расчета на графике. Проделать указанное для следующих вариантов начальных условий:

x_0	β	$M\varsigma$	t_0	R
$\begin{pmatrix} 0 \\ -1 \end{pmatrix}$	0.5	10	1	0.25
$\begin{pmatrix} -4 \\ 0 \end{pmatrix}$	0.5	10	0.75	0.15
$\begin{pmatrix} 0 \\ 3 \end{pmatrix}$	0.5	10	0.1	0.01

Для того, чтобы скрыть ненужный фрагмент текста программы, расположенный между исходными данными и графиком, необходимо воспользоваться алгоритмом действий, приведенным в параграфе «Скрыть фрагмент кода и ограничить доступ к нему в программном модуле в пакете MathCAD 15» на стр. 8.

1.4 Задачи для самостоятельного решения

Требования к выполнению самостоятельной работы:

Аналитический метод: найти стационарные точки, произвести их анализ и доказать, к какому типу относится каждая из стационарных точек (минимум, максимум или седловая точка).

Численный метод:

- рассчитать последовательность точек, используя построчную реализацию метода **случайного поиска с возвратом при неудачном шаге** (см. Пример 1.1 на стр. 13);

- подобрать начальные параметры таким образом, чтобы погрешность расчета составила: 10%, 25%, 50% по сравнению с теоретически полученным значением.

Вариацию параметров производить при расчете с помощью единой пользовательской функции. Для одного из подобранных вариантов начальных условий произвести построчную реализацию метода **случайного поиска с возвратом при неудачном шаге**;

- построить трехмерный графики: поверхности и линий уровня, нанести точки теоретически полученных экстремумов и последовательность расчетных точек.

Пример 1.4

Найти безусловный экстремум целевой функции $f(x) = x_1^3 + x_2^3 - 3 \cdot x_1 \cdot x_2$ **аналитическим методом, методом случайного поиска с возвратом при неудачном шаге, построить график функции и нанести расчетные точки.**
Координата начальной точки: $x0 = \begin{pmatrix} 3 & 3 \end{pmatrix}^T$

Ответ: $x = \begin{pmatrix} 1 & 1 \end{pmatrix}^T$ точка минимума.

Пример 1.5

Найти безусловный экстремум целевой функции $f(x) = x_1^2 + x_2^2$ **аналитическим методом, методом случайного поиска с возвратом при неудачном шаге, построить график функции и нанести расчетные точки.**
Координата начальной точки: $x0 = \begin{pmatrix} 3 & 3 \end{pmatrix}^T$

Ответ: $x = \begin{pmatrix} 0 & 0 \end{pmatrix}^T$ точка минимума.

Пример 1.6

Найти безусловный экстремум целевой функции $f(x) = x_1^2 - x_2^2$ **аналитическим методом, методом случайного поиска с возвратом при неудачном шаге, построить график функции и нанести расчетные точки.**
Координата начальной точки: $x0 = \begin{pmatrix} 3 & 0 \end{pmatrix}^T$

Ответ: $x = \begin{pmatrix} 0 & 0 \end{pmatrix}^T$ не является точкой минимума или максимума, а является седловой точкой.

Пример 1.7

Найти безусловный экстремум целевой функции $f(x) = \left(1 - x_1\right)^2 + 10\left(x_2 - x_1\right)^2$ **аналитическим методом, методом случайного поиска с возвратом при неудачном шаге, построить график функции и нанести расчетные точки.**
Координата начальной точки: $x0 = \begin{pmatrix} 3 & 3 \end{pmatrix}^T$

Ответ: $x = \begin{pmatrix} 1 & 1 \end{pmatrix}^T$ является точкой глобального минимума.

Пример 1.8

Найти безусловный экстремум целевой функции $f(x) = x_1^2 + x_2^4$ **аналитическим методом, методом случайного поиска с возвратом при неудачном шаге, построить график функции и нанести расчетные точки.**
Координата начальной точки: $x0 = \begin{pmatrix} 3 & 3 \end{pmatrix}^T$

Ответ: $x = \begin{pmatrix} 0 & 0 \end{pmatrix}^T$ является точкой глобального минимума.

Пример 1.9

Найти безусловный экстремум целевой функции $f(x) = -x_1^2 - x_2^2 - x_3^2 - x_1 + x_1x_2 + 2x_3$ **аналитическим методом и методом случайного поиска с возвратом при неудачном шаге.**
Координата начальной точки: $x0 = \begin{pmatrix} 3 & 3 & 1 \end{pmatrix}^T$

Ответ: $x = \begin{pmatrix} -\frac{2}{3} & -\frac{1}{3} & 1 \end{pmatrix}^T$ является точкой локального максимума.

Пример 1.10

Найти безусловный экстремум целевой функции $f(x) = 2x_1^3 + 4x_1x_2^2 - 10x_1x_2 + x_2^2$ **аналитическим методом, методом случайного поиска с возвратом при неудачном шаге, построить график функции и нанести расчетные точки.**
Координата начальной точки: $x0 = \begin{pmatrix} 3 & 3 \end{pmatrix}^T$

Ответ: в точке $x_1^* = \begin{pmatrix} 1 & 1 \end{pmatrix}^T$ – локальный минимум, в точке $x_2^* = \begin{pmatrix} 0 & 0 \end{pmatrix}^T$ – нет экстремума, так как не выполняются необходимые условия экстремума второго порядка.

Пример 1.11

Найти безусловный экстремум целевой функции $f(x) = \left(x_1 - 1\right)^4 + \left(x_2 - 3\right)^2$ **аналитическим методом, методом случайного поиска с возвратом при неудачном шаге, построить график функции и нанести расчетные точки.**
Координата начальной точки: $x0 = \begin{pmatrix} 3 & 3 \end{pmatrix}^T$

Ответ: в точке $x_1^* = \begin{pmatrix} 1 & 3 \end{pmatrix}^T$ – является точкой глобального минимума.

Пример 1.12

Найти безусловный экстремум целевой функции
$f(x)=x_1^3-x_1x_2+x_2^2-2x_1+3x_2-4$ **аналитическим методом, методом случайного поиска с возвратом при неудачном шаге, построить график функции и нанести расчетные точки.**
Координата начальной точки: $x0=\begin{pmatrix}3 & 3\end{pmatrix}^T$

Ответ: в точке $x_1^*=\begin{pmatrix}\frac{1}{2} & -\frac{5}{4}\end{pmatrix}^T$ – локальный минимум, в точке $x_1^*=\begin{pmatrix}-\frac{1}{3} & -\frac{5}{3}\end{pmatrix}^T$ – нет экстремума, так как не выполняются необходимые условия экстремума второго порядка.

Пример 1.13

Найти безусловный экстремум целевой функции $f(x)=\left(x_2-x_1^2\right)^2+\left(1-x_1\right)^2$ **аналитическим методом, методом случайного поиска с возвратом при неудачном шаге, построить график функции и нанести расчетные точки.**
Координата начальной точки: $x0=\begin{pmatrix}3 & 3\end{pmatrix}^T$

Ответ: в точке $x_1^*=\begin{pmatrix}1 & 1\end{pmatrix}^T$ – локальный минимум.

Пример 1.14

Найти безусловный экстремум целевой функции Хаммельблау
$f(x)=\left[x_0^2+x_1+11\right]^2+\left[x_0+x_1^2-7\right]^2$ **аналитическим методом, методом случайного поиска с возвратом при неудачном шаге, построить график функции и нанести расчетные точки.**
Координата начальной точки - $x0=\begin{pmatrix}3 & 3\end{pmatrix}^T$**. Подобрать координаты начальных точек, чтобы при реализации метода случайного поиска с возвратом при неудачном шаге были найдены другие (три) точки минимума.**

Ответ: в точках $x_1^*=\begin{pmatrix}-3.779 & -3.283\end{pmatrix}^T$, $x_2^*=\begin{pmatrix}-2.805 & 3.131\end{pmatrix}^T$, $x_3^*=\begin{pmatrix}3.584 & -1.848\end{pmatrix}^T$, $x_1^*=\begin{pmatrix}3 & 2\end{pmatrix}^T$ – локальный минимум.

Пример 1.15

Найти безусловный экстремум целевой функции $f(x) = 3x_1x_2 - x_1x_2^2 - x_1^2x_2$ **аналитическим методом, методом случайного поиска с возвратом при неудачном шаге, построить график функции и нанести расчетные точки.**
Координата начальной точки: $x0 = \begin{pmatrix} 3 & 3 \end{pmatrix}^T$

Ответ: в точке $x_1^* = \begin{pmatrix} 1 & 1 \end{pmatrix}^T$ – локальный минимум, в точке $x_2^* = \begin{pmatrix} 0 & 0 \end{pmatrix}^T$ – нет экстремума.

Пример 1.16

Найти безусловный максимум целевой функции Хаммельблау $f(x) = \left[x_0^2 + x_1 + 11\right]^2 + \left[x_0 + x_1^2 - 7\right]^2$ **аналитическим методом, методом случайного поиска с возвратом при неудачном шаге, построить график функции и нанести расчетные точки.**
Координата начальной точки: $x0 = \begin{pmatrix} 3 & 3 \end{pmatrix}^T$

Ответ: в точках $x_1^* = \begin{pmatrix} -0.271 & -0.923 \end{pmatrix}^T$ – локальный максимум.

Пример 1.17

Проверить, является ли точки $x_1^* = \begin{pmatrix} 0 & 0 \end{pmatrix}^T$**,** $x_2^* = \begin{pmatrix} 1 & 1 \end{pmatrix}^T$**,** $x_3^* = \begin{pmatrix} -1 & -1 \end{pmatrix}^T$ **точками безусловного минимума целевой функции** $f(x) = x_1^4 + x_2^4 - \left(x_1 + x_2\right)^2$ **аналитическим методом, методом случайного поиска с возвратом при неудачном шаге, построить график функции и нанести расчетные точки. Координата начальной точки -** $x0 = \begin{pmatrix} 3 & 3 \end{pmatrix}^T$**. Подобрать координаты начальной точки, чтобы при реализации метода случайного поиска с возвратом при неудачном шаге была найдена точка** $x_3^* = \begin{pmatrix} -1 & -1 \end{pmatrix}^T$**.**

Ответ: точка $x_1^* = \begin{pmatrix} 0 & 0 \end{pmatrix}^T$ не является точкой безусловного локального минимума, точки $x_2^* = \begin{pmatrix} 1 & 1 \end{pmatrix}^T$ и $x_3^* = \begin{pmatrix} -1 & -1 \end{pmatrix}^T$ является точкой безусловного локального минимума.

1.5 Контрольные вопросы с ответами

1. Метод случайного поиска с возвратом при неудачном шаге относится к оптимизационным методам …………… (вставить слово) порядка.

Ответ: нулевого.

2. Производные какого порядка используются в методе случайного поиска с возвратом при неудачном шаге? (выбрать вариант).

1. Первого
2. Второго
3. Третьего
4. Четвертого
5. Не используются

Ответ: 5. Не используются.

3. В методе случайного поиска с возвратом при неудачном шаге используются системы направлений $d_1, d_2, ..., d_n$?

1. Да	2. Нет	3. В некоторых случаях	4. Вопрос не корректный

Ответ: 2. Нет.

4. При реализации метода случайного поиска с возвратом при неудачном шаге вычисляется случайный вектор $\varsigma_j = (\varsigma_{j_1}, \ldots, \varsigma_{j_n})^T$. Значения элементов вектора ς_{j_i} случайно и равномерно распределены на некотором интервале. Выбрать интервал.

1. (0 –1)	2. (–1 –0)	3. (–1 –1)	4.(–2 –2)	5. Интервал не указан

Ответ: 3. (–1 –1).

5. При реализации метода случайного поиска с возвратом при неудачном шаге вычисляется случайный вектор $\varsigma_j = (\varsigma_{j_1}, \ldots, \varsigma_{j_n})^T$. Какое количество элементов содержится в векторе, если исследуется уравнение с n неизвестными. Выбрать из таблицы

1. $n-10$	2. $n+1$	3. $n/2$	4. n	5. Нет такого значения

Ответ: 4. n.

6. При реализации метода случайного поиска с возвратом при неудачном шаге вычислен случайный вектор $\varsigma = \begin{pmatrix} 0.6 \\ 0.8 \end{pmatrix}$. Рассчитать значение единичного случайного вектора.

Ответ: $\xi = \begin{pmatrix} 0.6 \\ 0.8 \end{pmatrix}$ (пояснения: $\xi = \begin{pmatrix} 0.6 \\ 0.8 \end{pmatrix}$, единичный случайный вектор рассчитывается как $\xi = \frac{\varsigma}{\|\varsigma\|} = \frac{\varsigma}{\sqrt{\varsigma_0^2 + \varsigma_1^2}}$, $\xi = \frac{\begin{pmatrix} 0.6 \\ 0.8 \end{pmatrix}}{\sqrt{0.6^2 + 0.8^2}} = \frac{\begin{pmatrix} 0.6 \\ 0.8 \end{pmatrix}}{1} = \begin{pmatrix} 0.6 \\ 0.8 \end{pmatrix}$).

7. При реализации метода случайного поиска с возвратом при неудачном шаге вычисляются координаты случайной точки y_j. Какое выражение используется для расчета?

1. $y_j = -x_k + t_k\xi_k$	2. $y_j = x_k - t_k\xi_k$	3. $y_j = x_k + t_k\xi_k$	4. нет такого выражения

Ответ: 3. $y_j = x_k + t_k\xi_k$.

8. Что бы шаг по i-ому направлению был удачен при использовании метода случайного поиска с возвратом при неудачном шаге необходимо, что бы выполнялось следующее неравенство ………… (выбрать вариант).

1. $f(y_j) > f(x_k)$	2. $f(y_j) < f(x_k)$	3. $f(y_j) < f(y_{j-1})$	4. Нет неравенства

Ответ: 2. $f(y_j) < f(x_k)$.

9. Что бы шаг по i-ому направлению был не удачен при использовании метода случайного поиска с возвратом при неудачном шаге необходимо, что бы выполнялось следующее неравенство …………… (выбрать вариант).

1. $f(y_j) > f(x_k)$	2. $f(y_j) < f(x_k)$	3. $f(y_j) < f(y_{j-1})$	4. Нет неравенства

Ответ: 1. $f(y_j) > f(x_k)$.

10. При поиске минимума функции $f(x) = 4(x_0 - 5)^2 + (x_1 - 6)^2$ методом случайного поиска с возвратом при неудачном шаге рассчитать координаты случайной точки (y_j) для $x_k = \begin{pmatrix} 7.2 \\ 8.3 \end{pmatrix}$, $t_k = 1$, $\xi_j = \frac{\varsigma_j}{|\varsigma_j|} = \begin{pmatrix} -1 \\ 0.06 \end{pmatrix}$.

Ответ: $y_j = \begin{pmatrix} 6.2 \\ 8.36 \end{pmatrix}$ (комментарий $y_j = x_k + t_k\xi_j = \begin{pmatrix} 7.2 \\ 8.3 \end{pmatrix} + 1\begin{pmatrix} -1 \\ 0.06 \end{pmatrix} = \begin{pmatrix} 6.2 \\ 8.36 \end{pmatrix}$).

11. Сколько случайных точек анализируются одновременно при каждой итерации при поиске минимума функции с использованием метода случайного поиска с возвратом при неудачном шаге ………… (выбрать вариант).

1. Больше 1	2. Две	3. Одна	4. Нет значения	5. Равное количеству переменных в уравнении	6. Вопрос не корректен

Ответ: 3. Одна.

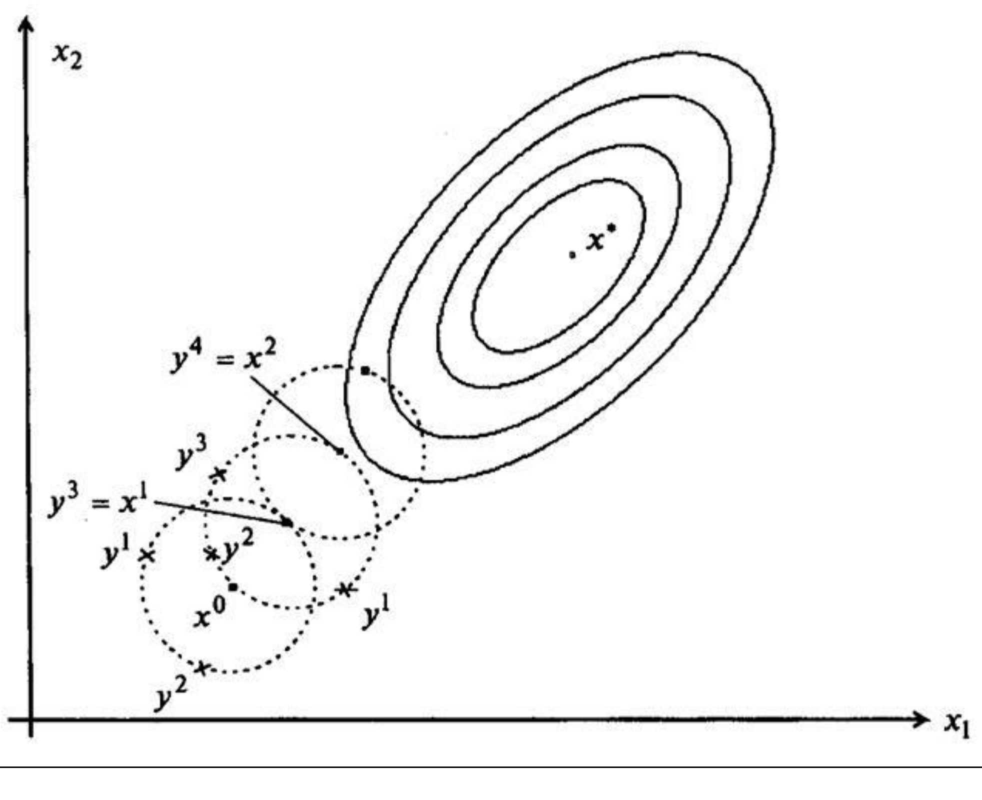
12. Если шаг по j-ому направлению был удачен при использовании метода случайного поиска с возвратом при неудачном шаге и выполняется условие $k = N$ (N – максимальное количество итераций с положительным шагом), то на следующей итерации модуль коэффициента β в выражении $t_k = \beta t_k$ для расчета длины шага должен быть ……… (выбрать вариант).

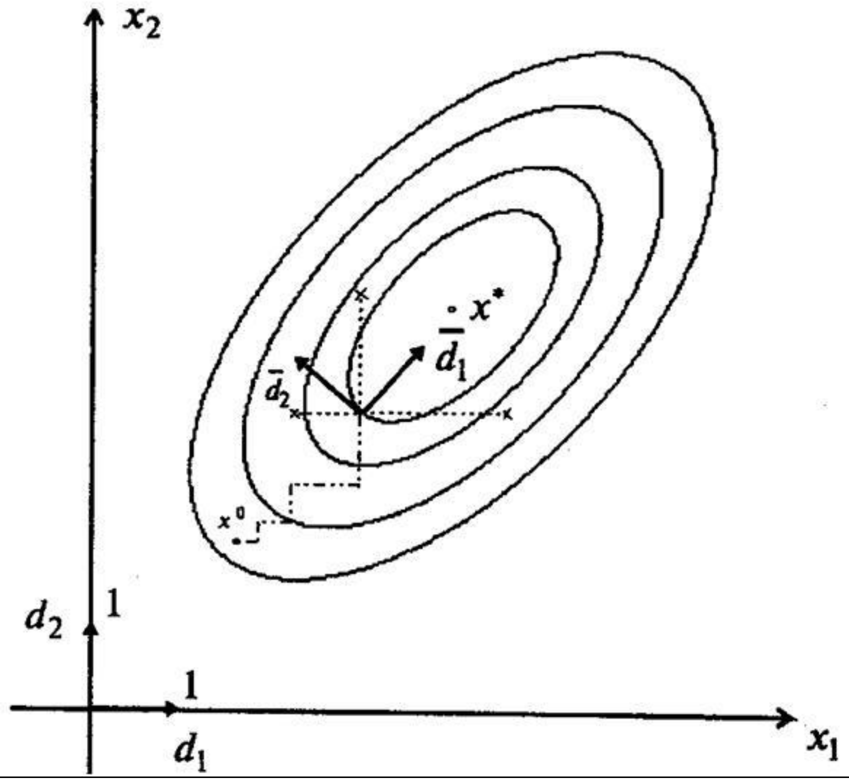
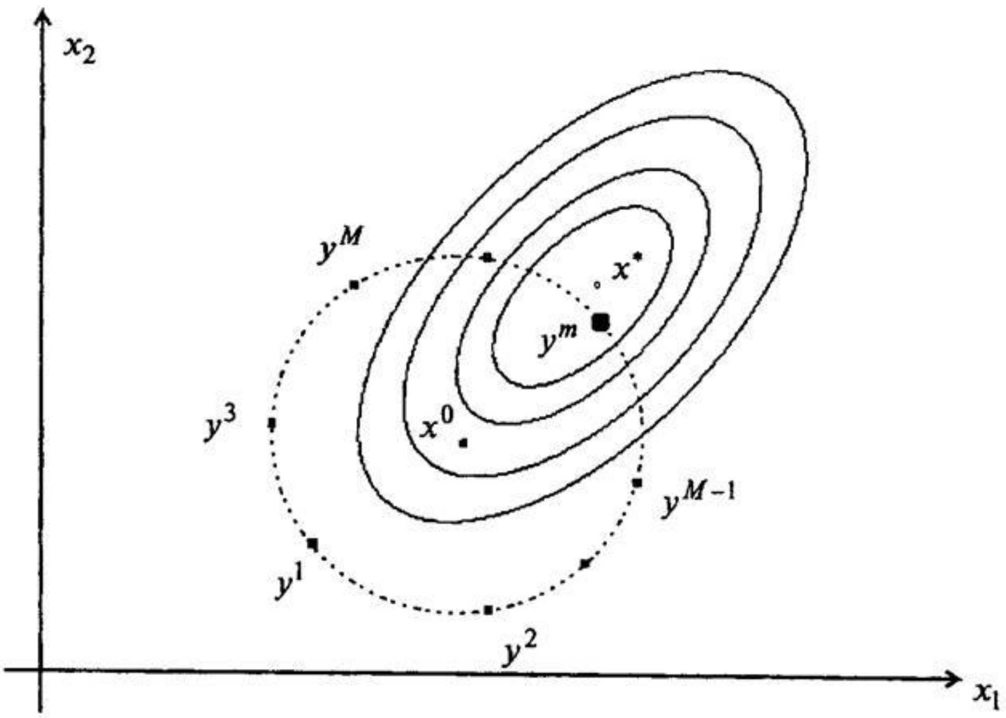
1. Больше 1	2. Меньше 1	3. Равен 1	4. Нет значения	5. Вопрос не корректен

Ответ: 5. Вопрос не корректен (комментарий: при таких условиях расчет завершен и следующей итерации нет).

13. Выбрать рисунок, который иллюстрирует работу метода случайного поиска с возвратом при неудачном шаге (выбрать вариант).

1.



2. 
3. 
4. Нет такого рисунка

Ответ: 1.

14. Если шаг по j-ому направлению был не удачен при использовании метода случайного поиска с возвратом при неудачном шаге и выполняется условие $j = M$ (M – максимальное количество неудач) и $t_k \le R$ (t_k – шаг, R – минимальная величина шага), то на следующей итерации модуль коэффициента β в выражении $t_k = \beta t_k$ для расчета длины шага должен быть ……… (выбрать вариант).

1. Больше 1	2. Меньше 1	3. Равен 1	4. Нет значения	5. Вопрос не корректен

Ответ: 5. Вопрос не корректен (комментарий: при таких условиях расчет завершен и следующей итерации нет).

15. Если после одной итерации при реализации метода случайного поиска с возвратом при неудачном шаге, выполняется неравенство $f(y_j) < f(x_k)$ (y_j – координата случайной точки, x_k – координата точки при удачном шаге). Какой шаг алгоритма следует далее? (выбрать вариант).

1. Завершить расчет.
2. Следующая итерация с существующими текущими параметрами $x_{k+1} = x_k$, $t_{k+1} = t_k$, $k = k$.
3. Следующая итерация с параметрами $x_{k+1} = y_j$, $t_{k+1} = t_k$, $k = k+1$.
4. Проверить окончание расчета по выполнению неравенства $t_k \le R$.
5. Проверяется число неудачных итераций.
6. Оценить число неудачных шагов из текущей центральной точки.

Ответ: 3. Следующая итерация с параметрами $x_{k+1} = y_j$, $t_{k+1} = t_k$, $k = k+1$.

16. Если после одной итерации при реализации метода случайного поиска с возвратом при неудачном шаге, выполняется неравенство $f(y_j) \ge f(x_k)$ (y_j – координата случайной точки, x_k – координата точки при удачном шаге). Какой шаг алгоритма следует далее? (выбрать вариант).

1. Завершить расчет.
2. Следующая итерация с существующими текущими параметрами $x_{k+1} = x_k$, $t_{k+1} = t_k$, $k = k$.
3. Следующая итерация с параметрами $x_{k+1} = y_j$, $t_{k+1} = t_k$, $k = k+1$.
4. Проверить окончание расчета по выполнению неравенства $t_k \le R$.
5. Проверяется число неудачных итераций.
6. Оценить число неудачных шагов из текущей центральной точки.

Ответ: 6. Оценить число неудачных шагов из текущей центральной точки.

17. Если шаг по j-ому направлению был не удачен при использовании метода случайного поиска с возвратом при неудачном шаге и выполняется условие $j < M$ (M – максимальное количество неудач) то на следующей итерации модуль коэффициента β в выражении $t_k = \beta t_k$ для расчета длины шага должен быть ……… (выбрать вариант).

1. Больше 1	2. Меньше 1	3. Рравен 1	4. Нет значения	5. Вопрос не корректен

Ответ: 3. Равен 1.

18. Если шаг по j-ому направлению был не удачен при использовании метода случайного поиска с возвратом при неудачном шаге и выполняется условие $j = M$ (M – максимальное количество неудач) и $t_k > R$ (t_k – шаг, R – минимальная величина шага), то на следующей итерации модуль коэффициента β в выражении $t_k = \beta t_k$ для расчета длины шага должен быть ……… (выбрать вариант).

1. Больше 1	2. Меньше 1	3. Равен 1	4. Нет значения	5. Вопрос не корректен

Ответ: 2. Меньше 1.

19. Если шаг по i-ому направлению был удачен при использовании метода случайного поиска с возвратом при неудачном шаге, то на следующей итерации модуль коэффициента β в выражении $t_k = \beta t_k$ для расчета длины шага для следующей итерации должен быть ……… (выбрать вариант).

1. Больше 1	2. Меньше 1	3. Равен 1	4. Нет значения

Ответ: 3. Равен 1.

20. При поиске минимума функции $f(x) = 4(x_0 - 5)^2 + (x_1 - 6)^2$ методом случайного поиска с возвратом при неудачном шаге рассчитать параметры для следующей итерации, если $y_j = \begin{pmatrix} 4.833 \\ 5.906 \end{pmatrix}$, $x_k = \begin{pmatrix} 5.183 \\ 6.262 \end{pmatrix}$, $t_k = 1$, коэффициент сжатия $\beta = 0.5$, $k = 6$.

Ответ: $x_{k+1} = \begin{pmatrix} 4.833 \\ 5.906 \end{pmatrix}$ $t_{k+1} = 1$, $k = 7$ (комментарий: так как

$f(y_j)$ =0.121< $f(x_k) = 0.203$ шаг удачен, положить $x_{k+1} = y_j = \begin{pmatrix} 4.833 \\ 5.906 \end{pmatrix}$, $t_{k+1} = t_k = 1$,

$k = k + 1 = 7$).

2 ЛИСТИНГИ ПРОГРАММ В ПАКЕТЕ MATHCAD 15

2.1 Листинги программы поиска минимума функции методом случайного поиска с возвратом при неудачном шаге в пакете MathCAD 15 (Пример 1.1)

Пример №1.1: Найти локальный минимум функции

$f(x) := 4\left(x_0 - 5\right)^2 + \left(x_1 - 6\right)^2$ методом случайного поиска с возвратом при не удачном шаге

Последовательность вычислений

Численный метод: реализовать алгоритм :

1. Построчно. Построить последовательность расчетных точек и рассчитать значение функции в этих точках, проверить выполнение условий, т.о. определить точку минимума.
2. С использованием одной пользовательской функции.
3. Сравнить результаты расчета

График. построить трехмерный график, нанести последовательность расчетных точек и точку локального минимума

Построить два типа графиков: 1. в виде поверхности. 2. в виде линий уровня

Вспомогательные функции и переменные, используемые при построчной реализации алгоритма

1. Функция для пересчета из блочного вектора в массив из двух колонок. На входе: xr- блочный вектор точек расчета.

На выходе: массив из двух колонок (x и y) - координат расчетных точек. Используется для подготовки данных расчета при построении графиков

```
v_k(xr) := | vx ← xr₀ᵀ
           | for  i ∈ 1..last(xr)
           |    vx ← stack(vx, xrᵢᵀ)
           | vx
```

Рис. 2.1. Листинг программы расчета минимума функции методом случайного поиска с возвратом при неудачном шаге (Пример 1.1, часть 1)

2. Переменные (шу и шн), используются для оценки успешности шага по случайному направлению с помощью оператора -

проверка_шага := if$\left(f\left(y_j\right) < f\left(x_k\right), шу, шн\right)$

шу := "Шаг удачен. x_k+1=y_j, t_k+1=t_k, k=k+1. Усл. окончания по N"

шн := "Шаг не удачен. Оценить число неудач - M (Шаг 5)."

3. Переменные (оконч_k_меньше_N и оконч_k_равно_N), используются для оценки успешности
окончания расчета по параметру N с помощью оператора -

условие_оконч_по_N := if(k < N ,k_меньше_N ,k_равно_N)

k_меньше_N := "Положить j=0. Перейти к шагу 2."

k_равно_N := "k=N. Расчет завершен. Точка минимума x*=xk"

4. Переменные (оконч_j_меньше_M и оконч_j_равно_M), используются для оценки успешности
окончания расчета по параметру M с помощью оператора -

условие_оконч_по_M := if(j < M ,j_меньше_M ,j_равно_M)

j_меньше_M := "Положить j=j+1. Перейти к шагу 2."

j_равно_M := "Проверить условие окончания по R"

5. Переменные (tk_меньше_R и tk_больше_R), используются для оценки успешности
окончания расчета по параметру R с помощью оператора -

условие_оконч_по_R := if$\left(t_k \le R, tk_меньше_R, tk_больше_R\right)$

tk_меньше_R := "Расчет завершен. Точка минимума x*=xk"

tk_больше_R := "Положить tk =β*tk, j=0, перейти к шагу 2"

Построчная реализация алгоритма метода случайного поиска с возвратом при неудачном шаге

Рис. 2.2. Листинг программы расчета минимума функции методом случайного поиска с возвратом при неудачном шаге (Пример 1.1, часть 2)

1^0. Исходные данные и описание переменных, использованных при построчной реализации алгоритма

$f(x) := 4(x_0 - 5)^2 + (x_1 - 6)^2$ - исследуемая функция

$x0 := \begin{pmatrix} 8 \\ 9 \end{pmatrix}$ - начальная точка

$\beta := 0.5$ - коэффициент сжатия *$0<\beta<1$*

$M := 3$ - максимальный индекс числа неудачно выполненных испытаний на текущей итерации (всего 4 неудачи, так как для первого элемента $j := 0$)

$N := 10$ - предельный индекс числа итераций при удачном шаге (всего 11 удачных шагов, так как для первого элемента $k := 0$)

$t0 := 1$ - начальная величина шага

$R := 0.5$ - минимальная величина шага

$k := 0$ - индекс итерации переменной (точки) x_k при удачном шаге

x - вектор с координатами точек при удачном шаге

ξ - случайный вектор единичной длины, определяющий направление шага

$j := 0$ - индекс итерации случайной переменной (точки) - y_j

y - вектор с координатами случайных точек при случайном выборе направления шага

Рис. 2.3. Листинг программы расчета минимума функции методом случайного поиска с возвратом при неудачном шаге (Пример 1.1, часть 3)

Внимание: В учебных целях значение для случайного вектора рассчитывается с помощью функции $\mathrm{runif}(2,-1,1)$ на каждом шаге, а затем копируется в переменную ζ и значение для ξ становится постоянным для данного примера. В противном случае не удается добиться стабильно работающего алгоритма для этого метода. При написании пользовательской функции, реализующий алгоритм, этого допущения нет.

Итерация $k = 0$ точек x_k

2^0 Расчет случайного вектора ξ при $j = 0$ итерации ξ_j

$\mathrm{runif}(\mathrm{last}(x0) + 1, -1, 1) = \begin{pmatrix} 0.567 \\ 0.04 \end{pmatrix}$ - расчет случайного вектора

равномерно распеределенный в итервале [-1 1] $\zeta := \begin{pmatrix} 0.646 \\ -0.652 \end{pmatrix}$ - случайный вектор

$\xi_j := \dfrac{\zeta}{|\zeta|}$, $\xi_j = \begin{pmatrix} 0.704 \\ -0.71 \end{pmatrix}$ - случайный вектор единичной длины

$|\xi_j| = 1$, определяющий направление шага

3^0 Расчет координаты случайного вектора.

Положим $t_k := t0$, $x_k := x0$ и $y_j := x_k + t_k \cdot \xi_j$, $y_j = \begin{pmatrix} 8.704 \\ 8.29 \end{pmatrix}$ - координаты случайной точки

4^0 Проверить выполнение условий: удачного шага и окончания расчета по M

Проверка уловия удачного шага

$f(x_k) = 45$ $\quad f(y_j) = 60.116$ сравниваемые величины

Рис. 2.4. Листинг программы расчета минимума функции методом случайного поиска с возвратом при неудачном шаге (Пример 1.1, часть 4)

$проверка\_шага := if(f(y_j) < f(x_k), шу, шн)$

$проверка\_шага =$ "Шаг не удачен. Оценить число неудач - M (Шаг 5)."

5^0 Оценить число локальных неудач по параметру M при расчете случайных точек

$j = 0$, $M = 3$ - контролируемые параметры

$условие\_оконч\_по\_M := if(j < M, j\_меньше\_M, j\_равно\_M)$

$условие\_оконч\_по\_M =$ "Положить j=j+1. Перейти к шагу 2."

$j := j + 1 = 1$

2^1 Расчет случайного вектора ξ при $j = 1$ итерации ξ_j

$runif(last(x0) + 1, -1, 1) = \begin{pmatrix} 0.752 \\ 0.912 \end{pmatrix}$ - расчет случайного вектора

равномерно распепеделенный в итервале [-1 1] $\zeta := \begin{pmatrix} 0.315 \\ 0.684 \end{pmatrix}$ - случайный вектор

$\xi_j := \dfrac{\zeta}{|\zeta|}$, $\xi_j = \begin{pmatrix} 0.418 \\ 0.908 \end{pmatrix}$ - случайный вектор единичной длины

$|\xi_j| = 1$, определяющий направление шага

3^1 Расчет координаты случайного вектора.

$t_k = 1$, $x_k = \begin{pmatrix} 8 \\ 9 \end{pmatrix}$ и $y_j := x_k + t_k \cdot \xi_j$, $y_j = \begin{pmatrix} 8.418 \\ 9.908 \end{pmatrix}$ - координаты случайной точки

4^1 Проверить выполнение условий: удачного шага и окончания расчета по M

Рис. 2.5. Листинг программы расчета минимума функции методом случайного поиска с возвратом при неудачном шаге (Пример 1.1, часть 5)

Проверка уловия удачного шага

$f(x_k) = 45 \quad f(y_j) = 62.014$ сравниваемые величины

$проверка\_шага := if(f(y_j) < f(x_k), шу, шн)$

$проверка\_шага =$ "Шаг не удачен. Оценить число неудач - М (Шаг 5)."

5^1 Оценить число локальных неудач по параметру М при расчете случайных точек

$j = 1$, $M = 3$ - контролируемые параметры

$условие\_оконч\_по\_M := if(j < M, j\_меньше\_M, j\_равно\_M)$

$условие\_оконч\_по\_M =$ "Положить j=j+1. Перейти к шагу 2."

$j := j + 1 = 2$

2^2 Расчет случайного вектора ξ при $j = 2$ итерации ξ_j

$runif(last(x0) + 1, -1, 1) = \begin{pmatrix} 0.079 \\ -0.076 \end{pmatrix}$ - расчет случайного вектора

равномерно распеределенный в итервале [-1 1] $\zeta := \begin{pmatrix} -0.817 \\ -0.705 \end{pmatrix}$ - случайный вектор

$\xi_j := \frac{\zeta}{|\zeta|}$, $\xi_j = \begin{pmatrix} -0.757 \\ -0.653 \end{pmatrix}$ - случайный вектор единичной длины $|\xi_j| = 1$, определяющий направление шага

3^2 Расчет координаты случайного вектора.

$t_k = 1$, $x_k = \begin{pmatrix} 8 \\ 9 \end{pmatrix}$ и $y_j := x_k + t_k \cdot \xi_j$, $y_j = \begin{pmatrix} 7.243 \\ 8.347 \end{pmatrix}$ - координаты случайной точки

Рис. 2.6. Листинг программы расчета минимума функции методом случайного поиска с возвратом при неудачном шаге (Пример 1.1, часть 6)

4^2 Проверить выполнение условий: удачного шага и окончания расчета по N

Проверка уловия удачного шага

$f(x_k) = 45 \quad f(y_j) = 25.629$ сравниваемые величины

проверка_шага := if(f(y_j) < f(x_k), шу, шн)

проверка_шага = "Шаг удачен. x_k+1=y_j, t_k+1=t_k, k=k+1. Усп. окончания по N"

Положить $x_{k+1} := y_j$, $x_{k+1} = \begin{pmatrix} 7.243 \\ 8.347 \end{pmatrix}$, $t_{k+1} := t_k$,

$k := k + 1 = 1$

Выполнение условия окончания расчета по N

$k = 1 \quad N = 10$ контролируемые параметры

условие_оконч_по_N := if(k < N, k_меньше_N, k_равно_N)

условие_оконч_по_N = "Положить j=0. Перейти к шагу 2."

$j := 0$

Итерация $k = 1$ точек x_k

2^3 Расчет случайного вектора ξ при $j = 0$ итерации ξ_j

$\mathrm{runif}(\mathrm{last}(x0) + 1, -1, 1) = \begin{pmatrix} 0.724 \\ 0.559 \end{pmatrix}$ - расчет случайного вектора

равномерно распеределенный в итервале [-1 1] $\zeta := \begin{pmatrix} -0.982 \\ 0.063 \end{pmatrix}$ - случайный вектор

Рис. 2.7. Листинг программы расчета минимума функции методом случайного поиска с возвратом при неудачном шаге (Пример 1.1, часть 7)

$\xi_j := \dfrac{\zeta}{|\zeta|}$, $\xi_j = \begin{pmatrix} -0.998 \\ 0.064 \end{pmatrix}$ - случайный вектор единичной длины

$|\xi_j| = 1$, определяющий направление шага

3^3 Расчет координаты случайного вектора.

$t_k = 1$, $x_k = \begin{pmatrix} 7.243 \\ 8.347 \end{pmatrix}$ и $y_j := x_k + t_k \cdot \xi_j$, $y_j = \begin{pmatrix} 6.245 \\ 8.411 \end{pmatrix}$ - координаты случайной точки

4^3 Проверить выполнение условий: удачного шага и окончания расчета по N

Проверка уловия удачного шага

$f(x_k) = 25.629$ $f(y_j) = 12.011$ сравниваемые величины

проверка_шага := if(f(y_j) < f(x_k), шу, шн)

проверка_шага = "Шаг удачен. x_k+1=y_j, t_k+1=t_k, k=k+1. Усл. окончания по N"

Положить $x_{k+1} := y_j$, $x_{k+1} = \begin{pmatrix} 6.245 \\ 8.411 \end{pmatrix}$,$t_{k+1} := t_k$,

$k := k + 1 = 2$

Выполнение условия окончания расчета по N

$k = 2$ $N = 10$ контролируемые параметры

условие_оконч_по_N := if(k < N, k_меньше_N, k_равно_N)

условие_оконч_по_N = "Положить j=0. Перейти к шагу 2."

$j := 0$

Рис. 2.8. Листинг программы расчета минимума функции методом случайного поиска с возвратом при неудачном шаге (Пример 1.1, часть 8)

Итерация $k = 2$ точек x_k

2^4 Расчет случайного вектора ξ при $j = 0$ итерации ξ_j

$\mathrm{runif}(\mathrm{last}(x0) + 1, -1, 1) = \begin{pmatrix} 0.994 \\ 0.223 \end{pmatrix}$ - расчет случайного вектора равномерно распеределенный в итервале [-1 1] $\zeta := \begin{pmatrix} -0.098 \\ -0.886 \end{pmatrix}$ - случайный вектор

$\xi_j := \dfrac{\zeta}{|\zeta|}$, $\xi_j = \begin{pmatrix} -0.11 \\ -0.994 \end{pmatrix}$ - случайный вектор единичной длины $|\xi_j| = 1$, определяющий направление шага

3^4 Расчет координаты случайного вектора.

$t_k = 1$, $x_k = \begin{pmatrix} 6.245 \\ 8.411 \end{pmatrix}$ и $y_j := x_k + t_k \cdot \xi_j$, $y_j = \begin{pmatrix} 6.135 \\ 7.417 \end{pmatrix}$ - координаты случайной точки

4^4 Проверить выполнение условий: удачного шага и окончания расчета по N

<u>Проверка уловия удачного шага</u>

$f(x_k) = 12.011 f(y_j) = 7.16$ сравниваемые величины

$\text{проверка\_шага} := \mathrm{if}(f(y_j) < f(x_k), \text{шу}, \text{шн})$

проверка_шага = "Шаг удачен. x_k+1=y_j, t_k+1=t_k, k=k+1. Усл. окончания по N"

Положить $x_{k+1} := y_j$, $x_{k+1} = \begin{pmatrix} 6.135 \\ 7.417 \end{pmatrix}$,$t_{k+1} := t_k$, $k := k + 1 = 3$

Рис. 2.9. Листинг программы расчета минимума функции методом случайного поиска с возвратом при неудачном шаге (Пример 1.1, часть 9)

Выполнение условия окончания расчета по N

$k = 3$ $N = 10$ контролируемые параметры

условие_оконч_по_N := if(k < N ,k_меньше_N ,k_равно_N)

условие_оконч_по_N = "Положить j=0. Перейти к шагу 2."

$j := 0$

Итерация $k = 3$ точек x_k

2^5 Расчет случайного вектора ξ при $j = 0$ итерации ξ_j

$\mathrm{runif}(\mathrm{last}(x0) + 1, -1, 1) = \begin{pmatrix} -0.468 \\ 0.68 \end{pmatrix}$ - расчет случайного вектора равномерно распеределенный в итервале [-1 1] $\zeta := \begin{pmatrix} 0.752 \\ 0.912 \end{pmatrix}$ - случайный вектор

$\xi_j := \frac{\zeta}{|\zeta|}$, $\xi_j = \begin{pmatrix} 0.636 \\ 0.772 \end{pmatrix}$ - случайный вектор единичной длины $|\xi_j| = 1$, определяющий направление шага

3^5 Расчет координаты случайного вектора.

$t_k = 1$, $x_k = \begin{pmatrix} 6.135 \\ 7.417 \end{pmatrix}$ и $y_j := x_k + t_k \cdot \xi_j$, $y_j = \begin{pmatrix} 6.771 \\ 8.188 \end{pmatrix}$ - координаты случайной точки

Рис. 2.10. Листинг программы расчета минимума функции методом случайного поиска с возвратом при неудачном шаге (Пример 1.1, часть 10)

4[5] Проверить выполнение условий: удачного шага и окончания расчета по N

Проверка уловия удачного шага

$f(x_k) = 7.16 \quad f(y_j) = 17.337$ сравниваемые величины

$проверка\_шага := if(f(y_j) < f(x_k), шу, шн)$

проверка_шага = "Шаг не удачен. Оценить число неудач - M (Шаг 5)."

5[2] Оценить число локальных неудач по параметру M при расчете случайных точек

$j = 0$, $M = 3$ - контролируемые параметры

$условие\_оконч\_по\_M := if(j < M, j\_меньше\_M, j\_равно\_M)$

условие_оконч_по_M = "Положить j=j+1. Перейти к шагу 2."

$j := j + 1 = 1$

2[6] Расчет случайного вектора ξ при $j = 1$ итерации ξ_j

$runif(last(x0) + 1, -1, 1) = \begin{pmatrix} -0.248 \\ 0.354 \end{pmatrix}$ - расчет случайного вектора

равномерно распепеделенный в итервале [-1 1] $\zeta := \begin{pmatrix} 0.724 \\ 0.559 \end{pmatrix}$ - случайный вектор

$\xi_j := \dfrac{\zeta}{|\zeta|}, \ \xi_j = \begin{pmatrix} 0.792 \\ 0.611 \end{pmatrix}$ - случайный вектор единичной длины

$|\xi_j| = 1$, определяющий направление шага

Рис. 2.11. Листинг программы расчета минимума функции методом случайного поиска с возвратом при неудачном шаге (Пример 1.1, часть 11)

3^6 Расчет координаты случайного вектора.

$t_k = 1$, $x_k = \begin{pmatrix} 6.135 \\ 7.417 \end{pmatrix}$ и $y_j := x_k + t_k \cdot \xi_j$, $y_j = \begin{pmatrix} 6.927 \\ 8.028 \end{pmatrix}$ - координаты случайной точки

4^6 Проверить выполнение условий: удачного шага и окончания расчета по М

<u>Проверка уловия удачного шага</u>

$f(x_k) = 7.16$ $f(y_j) = 18.959$ сравниваемые величины

проверка_шага := if(f(y_j) < f(x_k) ,шу ,шн)

проверка_шага = "Шаг не удачен. Оценить число неудач - М (Шаг 5)."

5^3 Оценить число локальных неудач по параметру М при расчете случайных точек

$j = 1$, $M = 3$ - контролируемые параметры

условие_оконч_по_М := if(j < M ,j_меньше_М ,j_равно_М)

условие_оконч_по_М = "Положить j=j+1. Перейти к шагу 2."

$j := j + 1 = 2$

2^7 Расчет случайного вектора ξ при $j = 2$ итерации ξ_j

$\mathrm{runif}(\mathrm{last}(x0) + 1, -1, 1) = \begin{pmatrix} -0.982 \\ -0.448 \end{pmatrix}$ - расчет случайного вектора равномерно распеределенный в итервале [-1 1] $\zeta := \begin{pmatrix} 0.17 \\ -0.299 \end{pmatrix}$ - случайный вектор

Рис. 2.12. Листинг программы расчета минимума функции методом случайного поиска с возвратом при неудачном шаге (Пример 1.1, часть 12)

$\xi_j := \frac{\zeta}{|\zeta|}$, $\xi_j = \begin{pmatrix} 0.494 \\ -0.869 \end{pmatrix}$ - случайный вектор единичной длины

$|\xi_j| = 1$, определяющий направление шага

3[7] Расчет координаты случайного вектора.

$t_k = 1$, $x_k = \begin{pmatrix} 6.135 \\ 7.417 \end{pmatrix}$ и $y_j := x_k + t_k \cdot \xi_j$, $y_j = \begin{pmatrix} 6.629 \\ 6.547 \end{pmatrix}$ - координаты случайной точки

4[7] Проверить выполнение условий: удачного шага и окончания расчета по М

Проверка уловия удачного шага

$f(x_k) = 7.16$ $f(y_j) = 10.918$ сравниваемые величины

проверка_шага := if(f(y_j) < f(x_k), шу, шн)

проверка_шага = "Шаг не удачен. Оценить число неудач - М (Шаг 5)."

5[4] Оценить число локальных неудач по параметру М при расчете случайных точек

$j = 2$, $M = 3$ - контролируемые параметры

условие_оконч_по_М := if(j < M, j_меньше_М, j_равно_М)

условие_оконч_по_М = "Положить j=j+1. Перейти к шагу 2."

$j := j + 1 = 3$

2[8] Расчет случайного вектора ξ при $j = 3$ итерации ξ_j

runif(last(x0) + 1, −1, 1) = $\begin{pmatrix} 0.176 \\ 0.675 \end{pmatrix}$ - расчет случайного вектора

равномерно распеределенный в итервале [-1 1] $\zeta := \begin{pmatrix} -0.817 \\ -0.705 \end{pmatrix}$ - случайный вектор

Рис. 2.13. Листинг программы расчета минимума функции методом случайного поиска с возвратом при неудачном шаге (Пример 1.1, часть 13)

$\xi_j := \frac{\zeta}{|\zeta|}$, $\xi_j = \begin{pmatrix} -0.757 \\ -0.653 \end{pmatrix}$ - случайный вектор единичной длины

$|\xi_j| = 1$, определяющий направление шага

3^8 Расчет координаты случайного вектора.

$t_k = 1$, $x_k = \begin{pmatrix} 6.135 \\ 7.417 \end{pmatrix}$ и $y_j := x_k + t_k \cdot \xi_j$, $y_j = \begin{pmatrix} 5.378 \\ 6.763 \end{pmatrix}$ - координаты случайной точки

4^8 Проверить выполнение условий: удачного шага и окончания расчета по N

<u>Проверка уловия удачного шага</u>

$f(x_k) = 7.16$ $f(y_j) = 1.154$ сравниваемые величины

проверка_шага := if(f(y_j) < f(x_k), шу, шн)

проверка_шага = "Шаг удачен. x_k+1=y_j, t_k+1=t_k, k=k+1. Усл. окончания по N"

Положить $x_{k+1} := y_j$, $x_{k+1} = \begin{pmatrix} 5.378 \\ 6.763 \end{pmatrix}$, $t_{k+1} := t_k$,

$k := k + 1 = 4$

<u>Выполнение условия окончания расчета по N</u>

$k = 4$ $N = 10$ контролируемые параметры

условие_оконч_по_N := if(k < N, k_меньше_N, k_равно_N)

условие_оконч_по_N = "Положить j=0. Перейти к шагу 2."

$j := 0$

Рис. 2.14. Листинг программы расчета минимума функции методом случайного поиска с возвратом при неудачном шаге (Пример 1.1, часть 14)

Итерация $k = 4$ точек x_k

2^9 Расчет случайного вектора ξ при $j = 0$ итерации ξ_j

$\mathrm{runif}(\mathrm{last}(x0) + 1, -1, 1) = \begin{pmatrix} -0.03 \\ 0.487 \end{pmatrix}$ - расчет случайного вектора равномерно распеределенный в итервале [-1 1] $\zeta := \begin{pmatrix} -0.982 \\ 0.063 \end{pmatrix}$ - случайный вектор

$\xi_j := \frac{\zeta}{|\zeta|}$, $\xi_j = \begin{pmatrix} -0.998 \\ 0.064 \end{pmatrix}$ - случайный вектор единичной длины $|\xi_j| = 1$, определяющий направление шага

3^9 Расчет координаты случайного вектора.

$t_k = 1$, $x_k = \begin{pmatrix} 5.378 \\ 6.763 \end{pmatrix}$ и $y_j := x_k + t_k \cdot \xi_j$, $y_j = \begin{pmatrix} 4.38 \\ 6.827 \end{pmatrix}$ - координаты случайной точки

4^9 Проверить выполнение условий: удачного шага и окончания расчета по М

<u>Проверка уловия удачного шага</u>

$f(x_k) = 1.154$ $f(y_j) = 2.222$ сравниваемые величины

проверка_шага := if(f(y_j) < f(x_k), шу, шн)

проверка_шага = "Шаг не удачен. Оценить число неудач - М (Шаг 5)."

Рис. 2.15. Листинг программы расчета минимума функции методом случайного поиска с возвратом при неудачном шаге (Пример 1.1, часть 15)

5^5 Оценить число локальных неудач по параметру М при расчете случайных точек

$j = 0$, $M = 3$ - контролируемые параметры

условие_оконч_по_М := if(j < M ,j_меньше_M ,j_равно_M)

условие_оконч_по_М = "Положить j=j+1. Перейти к шагу 2."

$j := j + 1 = 1$

2^9 Расчет случайного вектора ξ при $j = 1$ итерации ξ_j

$\mathrm{runif}(\mathrm{last}(x0) + 1, -1, 1) = \begin{pmatrix} -0.084 \\ 0.489 \end{pmatrix}$ - расчет случайного вектора

равномерно распепеделенный в итервале [-1 1] $\zeta := \begin{pmatrix} -0.098 \\ -0.886 \end{pmatrix}$ - случайный вектор

$\xi_j := \dfrac{\zeta}{|\zeta|}$, $\xi_j = \begin{pmatrix} -0.11 \\ -0.994 \end{pmatrix}$ - случайный вектор единичной длины

$|\xi_j| = 1$, определяющий направление шага

3^{10} Расчет координаты случайного вектора.

$t_k = 1$, $x_k = \begin{pmatrix} 5.378 \\ 6.763 \end{pmatrix}$ и $y_j := x_k + t_k \cdot \xi_j$, $y_j = \begin{pmatrix} 5.268 \\ 5.77 \end{pmatrix}$ - координаты случайной точки

4^{10} Проверить выполнение условий: удачного шага и окончания расчета по N

Проверка уловия удачного шага

$f(x_k) = 1.154$ $f(y_j) = 0.34$ сравниваемые величины

Рис. 2.16. Листинг программы расчета минимума функции методом случайного поиска с возвратом при неудачном шаге (Пример 1.1, часть 16)

проверка_шага := if(f(y_j) < f(x_k), шу, шн)

проверка_шага = "Шаг удачен. x_k+1=y_j, t_k+1=t_k, k=k+1. Усл. окончания по N"

Положить $x_{k+1} := y_j$, $x_{k+1} = \begin{pmatrix} 5.268 \\ 5.77 \end{pmatrix}$, $t_{k+1} := t_k$,

$k := k + 1 = 5$

Выполнение условия окончания расчета по N

$k = 5$ $N = 10$ контролируемые параметры

условие_оконч_по_N := if(k < N, k_меньше_N, k_равно_N)

условие_оконч_по_N = "Положить j=0. Перейти к шагу 2."

$j := 0$

Итерация $k = 5$ точек x_k

2[11] Расчет случайного вектора ξ при $j = 0$ итерации ξ_j

$\mathrm{runif}(\mathrm{last}(x0) + 1, -1, 1) = \begin{pmatrix} 0.198 \\ 0.47 \end{pmatrix}$ - расчет случайного вектора равномерно распеределенный в итервале [-1 1] $\zeta := \begin{pmatrix} 0.752 \\ 0.912 \end{pmatrix}$ - случайный вектор

$\xi_j := \dfrac{\zeta}{|\zeta|}$, $\xi_j = \begin{pmatrix} 0.636 \\ 0.772 \end{pmatrix}$ - случайный вектор единичной длины $|\xi_j| = 1$, определяющий направление шага

Рис. 2.17. Листинг программы расчета минимума функции методом случайного поиска с возвратом при неудачном шаге (Пример 1.1, часть 17)

3[11] Расчет координаты случайного вектора.

$t_k = 1$, $x_k = \begin{pmatrix} 5.268 \\ 5.77 \end{pmatrix}$ и $y_j := x_k + t_k \cdot \xi_j$, $y_j = \begin{pmatrix} 5.904 \\ 6.541 \end{pmatrix}$ - координаты случайной точки

4[11] Проверить выполнение условий: удачного шага и окончания расчета по М

Проверка уловия удачного шага

$f(x_k) = 0.34$ $f(y_j) = 3.563$ сравниваемые величины

проверка_шага := if(f(y_j) < f(x_k), шу, шн)

проверка_шага = "Шаг не удачен. Оценить число неудач - М (Шаг 5)."

5[6] Оценить число локальных неудач по параметру М при расчете случайных точек

$j = 0$, $M = 3$ - контролируемые параметры

условие_оконч_по_М := if(j < M, j_меньше_М, j_равно_М)

условие_оконч_по_М = "Положить j=j+1. Перейти к шагу 2."

$j := j + 1 = 1$

2[12] Расчет случайного вектора ξ при $j = 1$ итерации ξ_j

$\mathrm{runif}(\mathrm{last}(x0) + 1, -1, 1) = \begin{pmatrix} 0.145 \\ -0.697 \end{pmatrix}$ - расчет случайного вектора равномерно распепеделенный в итервале [-1 1] $\zeta := \begin{pmatrix} 0.994 \\ 0.223 \end{pmatrix}$ - случайный вектор

Рис. 2.18. Листинг программы расчета минимума функции методом случайного поиска с возвратом при неудачном шаге (Пример 1.1, часть 18)

$\xi_j := \frac{\zeta}{|\zeta|}$, $\xi_j = \begin{pmatrix} 0.976 \\ 0.219 \end{pmatrix}$ - случайный вектор единичной длины $|\xi_j| = 1$, определяющий направление шага

3[12] Расчет координаты случайного вектора.

$t_k = 1$, $x_k = \begin{pmatrix} 5.268 \\ 5.77 \end{pmatrix}$ и $y_j := x_k + t_k \cdot \xi_j$, $y_j = \begin{pmatrix} 6.244 \\ 5.988 \end{pmatrix}$ - координаты случайной точки

4[12] Проверить выполнение условий: удачного шага и окончания расчета по M

Проверка уловия удачного шага

$f(x_k) = 0.34$ $f(y_j) = 6.188$ сравниваемые величины

проверка_шага := if(f(y_j) < f(x_k), шу, шн)

проверка_шага = "Шаг не удачен. Оценить число неудач - M (Шаг 5)."

5[7] Оценить число локальных неудач по параметру M при расчете случайных точек

$j = 1$, $M = 3$ - контролируемые параметры

условие_оконч_по_M := if(j < M, j_меньше_M, j_равно_M)

условие_оконч_по_M = "Положить j=j+1. Перейти к шагу 2."

$j := j + 1 = 2$

2[13] Расчет случайного вектора ξ при $j = 2$ итерации ξ_j

Рис. 2.19. Листинг программы расчета минимума функции методом случайного поиска с возвратом при неудачном шаге (Пример 1.1, часть 19)

$\mathrm{runif}(\mathrm{last}(x0)+1,-1,1)=\begin{pmatrix}-0.15\\0.034\end{pmatrix}$ - расчет случайного вектора равномерно распеределенный в итервале [-1 1] $\zeta := \begin{pmatrix}-0.248\\0.354\end{pmatrix}$ - случайный вектор

$\xi_j := \frac{\zeta}{|\zeta|}$, $\xi_j = \begin{pmatrix}-0.574\\0.819\end{pmatrix}$ - случайный вектор единичной длины $|\xi_j| = 1$, определяющий направление шага

3[13] Расчет координаты случайного вектора.

$t_k = 1$, $x_k = \begin{pmatrix}5.268\\5.77\end{pmatrix}$ и $y_j := x_k + t_k\cdot\xi_j$, $y_j = \begin{pmatrix}4.694\\6.589\end{pmatrix}$ - координаты случайной точки

4[13] Проверить выполнение условий: удачного шага и окончания расчета по М

Проверка уловия удачного шага

$f(x_k) = 0.34$ $f(y_j) = 0.72$ сравниваемые величины

проверка_шага := if(f(y_j) < f(x_k) ,шу ,шн)

проверка_шага = "Шаг не удачен. Оценить число неудач - М (Шаг 5)."

5[8] Оценить число локальных неудач по параметру М при расчете случайных точек

$j = 2$, $M = 3$ - контролируемые параметры

условие_оконч_по_M := if(j < M ,j_меньше_M ,j_равно_M)

условие_оконч_по_M = "Положить j=j+1. Перейти к шагу 2."

$j := j + 1 = 3$

Рис. 2.20. Листинг программы расчета минимума функции методом случайного поиска с возвратом при неудачном шаге (Пример 1.1, часть 20)

2^{14} Расчет случайного вектора ξ при $j = 3$ итерации ξ_j

$\mathrm{runif}(\mathrm{last}(x0) + 1, -1, 1) = \begin{pmatrix} 0.503 \\ -0.662 \end{pmatrix}$ - расчет случайного вектора

равномерно распеределенный в итервале [-1 1] $\zeta := \begin{pmatrix} 0.176 \\ 0.675 \end{pmatrix}$ - случайный вектор

$\xi_j := \dfrac{\zeta}{|\zeta|}$, $\xi_j = \begin{pmatrix} 0.252 \\ 0.968 \end{pmatrix}$ - случайный вектор единичной длины

$|\xi_j| = 1$, определяющий направление шага

3^{14} Расчет координаты случайного вектора.

$t_k = 1$, $x_k = \begin{pmatrix} 5.268 \\ 5.77 \end{pmatrix}$ и $y_j := x_k + t_k \cdot \xi_j$, $y_j = \begin{pmatrix} 5.52 \\ 6.737 \end{pmatrix}$ - координаты случайной точки

4^{14} Проверить выполнение условий: удачного шага и окончания расчета по М

<u>Проверка уловия удачного шага</u>

$f(x_k) = 0.34$ $f(y_j) = 1.626$ сравниваемые величины

проверка_шага := if(f(y_j) < f(x_k), шу, шн)

проверка_шага = "Шаг не удачен. Оценить число неудач - М (Шаг 5)."

5^{9} Оценить число локальных неудач по параметру М при расчете случайных точек

$j = 3$, $M = 3$ - контролируемые параметры

Рис. 2.21. Листинг программы расчета минимума функции методом случайного поиска с возвратом при неудачном шаге (Пример 1.1, часть 21)

условие_оконч_по_M := if(j < M, j_меньше_M, j_равно_M)

условие_оконч_по_M = "Проверить условие окончания по R"

$j := j + 1 = 4$

Проверить окончание расчета по минимальной величине шага

условие_оконч_по_R := if($t_k \le R$, tk_меньше_R, tk_больше_R)

условие_оконч_по_R = "Положить tk =β*tk, j=0, перейти к шагу 2"

$t_k := \beta \cdot t_k \quad t_k = 0.5 \qquad j := 0$

2[15] Расчет случайного вектора ξ при $j = 0$ итерации ξ_j

runif(last(x0) + 1, −1, 1) = $\begin{pmatrix} -0.016 \\ 0.4 \end{pmatrix}$ - расчет случайного вектора равномерно распепеделенный в итервале [-1 1] $\zeta := \begin{pmatrix} -0.705 \\ -0.717 \end{pmatrix}$ - случайный вектор

$\xi_j := \frac{\zeta}{|\zeta|}, \ \xi_j = \begin{pmatrix} -0.701 \\ -0.713 \end{pmatrix}$ - случайный вектор единичной длины $|\xi_j| = 1$, определяющий направление шага

3[15] Расчет координаты случайного вектора.

$t_k = 0.5$, $x_k = \begin{pmatrix} 5.268 \\ 5.77 \end{pmatrix}$ и $y_j := x_k + t_k \cdot \xi_j, \ y_j = \begin{pmatrix} 4.917 \\ 5.413 \end{pmatrix}$ - координаты случайной точки

Рис. 2.22. Листинг программы расчета минимума функции методом случайного поиска с возвратом при неудачном шаге (Пример 1.1, часть 22)

4^{15} Проверить выполнение условий: удачного шага и окончания расчета по М

<u>Проверка уловия удачного шага</u>

$f(x_k) = 0.34 \quad f(y_j) = 0.372$ сравниваемые величины

$проверка\_шага := if(f(y_j) < f(x_k), шу, шн)$

проверка_шага = "Шаг не удачен. Оценить число неудач - М (Шаг 5)."

5^{10} Оценить число локальных неудач по параметру М при расчете случайных точек

$j = 0$, $M = 3$ - контролируемые параметры

$условие\_оконч\_по\_M := if(j < M, j\_меньше\_M, j\_равно\_M)$

условие_оконч_по_М = "Положить j=j+1. Перейти к шагу 2."

$j := j + 1 = 1$

2^{16} Расчет случайного вектора ξ при $j = 1$ итерации ξ_j

$runif(last(x0) + 1, -1, 1) = \begin{pmatrix} -0.705 \\ -0.717 \end{pmatrix}$ - расчет случайного вектора равномерно распеределенный в итервале [-1 1] $\quad \zeta := \begin{pmatrix} 0.933 \\ -0.693 \end{pmatrix}$ - случайный вектор

$\xi_j := \dfrac{\zeta}{|\zeta|}, \ \xi_j = \begin{pmatrix} 0.803 \\ -0.596 \end{pmatrix}$ - случайный вектор единичной длины $|\xi_j| = 1$, определяющий направление шага

3^{16} Расчет координаты случайного вектора.

Рис. 2.23. Листинг программы расчета минимума функции методом случайного поиска с возвратом при неудачном шаге (Пример 1.1, часть 23)

$t_k = 0.5$, $x_k = \begin{pmatrix} 5.268 \\ 5.77 \end{pmatrix}$ и $y_j := x_k + t_k \cdot \xi_j$, $y_j = \begin{pmatrix} 5.669 \\ 5.471 \end{pmatrix}$ - координаты случайной точки

4^{16} Проверить выполнение условий: удачного шага и окончания расчета по М

Проверка уловия удачного шага

$f(x_k) = 0.34 \quad f(y_j) = 2.072$ сравниваемые величины

проверка_шага := if(f(y_j) < f(x_k), шу, шн)

проверка_шага = "Шаг не удачен. Оценить число неудач - М (Шаг 5)."

5^{11} Оценить число локальных неудач по параметру М при расчете случайных точек

$j = 1$, $M = 3$ - контролируемые параметры

условие_оконч_по_М := if(j < M, j_меньше_М, j_равно_М)

условие_оконч_по_М = "Положить j=j+1. Перейти к шагу 2."

$j := j + 1 = 2$

2^{17} Расчет случайного вектора ξ при $j = 2$ итерации ξ_j

$\mathrm{runif}(\mathrm{last}(x0) + 1, -1, 1) = \begin{pmatrix} 0.386 \\ -0.147 \end{pmatrix}$ - расчет случайного вектора равномерно распеределенный в итервале [-1 1] $\zeta := \begin{pmatrix} 0.464 \\ -0.441 \end{pmatrix}$ - случайный вектор

Рис. 2.24. Листинг программы расчета минимума функции методом случайного поиска с возвратом при неудачном шаге (Пример 1.1, часть 24)

$\xi_j := \frac{\zeta}{|\zeta|}$, $\xi_j = \begin{pmatrix} 0.725 \\ -0.689 \end{pmatrix}$ - случайный вектор единичной длины $|\xi_j| = 1$, определяющий направление шага

3[17] Расчет координаты случайного вектора.

$t_k = 0.5$, $x_k = \begin{pmatrix} 5.268 \\ 5.77 \end{pmatrix}$ и $y_j := x_k + t_k \cdot \xi_j$, $y_j = \begin{pmatrix} 5.63 \\ 5.425 \end{pmatrix}$ - координаты случайной точки

4[17] Проверить выполнение условий: удачного шага и окончания расчета по М

Проверка уловия удачного шага

$f(x_k) = 0.34$ $f(y_j) = 1.92$ сравниваемые величины

проверка_шага := if(f(y_j) < f(x_k) ,шу ,шн)

проверка_шага = "Шаг не удачен. Оценить число неудач - М (Шаг 5)."

5[12] Оценить число локальных неудач по параметру М при расчете случайных точек

$j = 2$, $M = 3$ - контролируемые параметры

условие_оконч_по_М := if(j < M ,j_меньше_М ,j_равно_М)

условие_оконч_по_М = "Положить j=j+1. Перейти к шагу 2."

$j := j + 1 = 3$

2[18] Расчет случайного вектора ξ при $j = 3$ итерации ξ_j

Рис. 2.25. Листинг программы расчета минимума функции методом случайного поиска с возвратом при неудачном шаге (Пример 1.1, часть 25)

$\text{runif}(\text{last}(x0) + 1, -1, 1) = \begin{pmatrix} 0.933 \\ -0.693 \end{pmatrix}$ - расчет случайного вектора равномерно распеределенный в итервале [-1 1] $\zeta := \begin{pmatrix} -0.084 \\ 0.489 \end{pmatrix}$ - случайный вектор

$\xi_j := \dfrac{\zeta}{|\zeta|}$, $\xi_j = \begin{pmatrix} -0.169 \\ 0.986 \end{pmatrix}$ - случайный вектор единичной длины $|\xi_j| = 1$, определяющий направление шага

3^{18} Расчет координаты случайного вектора.

$t_k = 0.5$, $x_k = \begin{pmatrix} 5.268 \\ 5.77 \end{pmatrix}$ и $y_j := x_k + t_k \cdot \xi_j$, $y_j = \begin{pmatrix} 5.183 \\ 6.262 \end{pmatrix}$ - координаты случайной точки

4^{18} Проверить выполнение условий: удачного шага и окончания расчета по N

Проверка уловия удачного шага

$f(x_k) = 0.34$ $f(y_j) = 0.203$ сравниваемые величины

$\text{проверка\_шага} := \text{if}(f(y_j) < f(x_k), \text{шу}, \text{шн})$

проверка_шага = "Шаг удачен. x_k+1=y_j, t_k+1=t_k, k=k+1. Усл. окончания по N"

Положить $x_{k+1} := y_j$, $x_{k+1} = \begin{pmatrix} 5.183 \\ 6.262 \end{pmatrix}$,$t_{k+1} := t_k$,

$k := k + 1 = 6$

Выполнение условия окончания расчета по N

Рис. 2.26. Листинг программы расчета минимума функции методом случайного поиска с возвратом при неудачном шаге (Пример 1.1, часть 26)

$k = 6 \qquad N = 10$ контролируемые параметры

условие_оконч_по_N := if(k < N, k_меньше_N, k_равно_N)

условие_оконч_по_N = "Положить j=0. Перейти к шагу 2."

$j := 0$

Итерация $k = 6$ точек x_k

2[19] Расчет случайного вектора ξ при $j = 0$ итерации ξ_j

$\mathrm{runif}(\mathrm{last}(x0) + 1, -1, 1) = \begin{pmatrix} 0.643 \\ -0.617 \end{pmatrix}$ - расчет случайного вектора равномерно распеределенный в итервале [-1 1] $\zeta := \begin{pmatrix} 0.145 \\ -0.697 \end{pmatrix}$ - случайный вектор

$\xi_j := \dfrac{\zeta}{|\zeta|}, \ \xi_j = \begin{pmatrix} 0.204 \\ -0.979 \end{pmatrix}$ - случайный вектор единичной длины $|\xi_j| = 1$, определяющий направление шага

3[19] Расчет координаты случайного вектора.

$t_k = 0.5$, $x_k = \begin{pmatrix} 5.183 \\ 6.262 \end{pmatrix}$ и $y_j := x_k + t_k \cdot \xi_j, \ y_j = \begin{pmatrix} 5.285 \\ 5.773 \end{pmatrix}$ - координаты случайной точки

4[19] Проверить выполнение условий: удачного шага и окончания расчета по M

Рис. 2.27. Листинг программы расчета минимума функции методом случайного поиска с возвратом при неудачном шаге (Пример 1.1, часть 27)

Проверка уловия удачного шага

$f(x_k) = 0.203 \quad f(y_j) = 0.377$ сравниваемые величины

проверка_шага := if(f(y_j) < f(x_k), шу, шн)

проверка_шага = "Шаг не удачен. Оценить число неудач - M (Шаг 5)."

5^{13} Оценить число локальных неудач по параметру M при расчете случайных точек

$j = 0$, $M = 3$ - контролируемые параметры

условие_оконч_по_M := if(j < M, j_меньше_M, j_равно_M)

условие_оконч_по_M = "Положить j=j+1. Перейти к шагу 2."

$j := j + 1 = 1$

2^{20} Расчет случайного вектора ξ при $j = 1$ итерации ξ_j

$\text{runif}(\text{last}(x0) + 1, -1, 1) = \begin{pmatrix} 0.634 \\ -0.689 \end{pmatrix}$ - расчет случайного вектора равномерно распеределенный в итервале [-1 1] $\zeta := \begin{pmatrix} -0.705 \\ -0.717 \end{pmatrix}$ - случайный вектор

$\xi_j := \dfrac{\zeta}{|\zeta|}, \quad \xi_j = \begin{pmatrix} -0.701 \\ -0.713 \end{pmatrix}$ - случайный вектор единичной длины $|\xi_j| = 1$, определяющий направление шага

3^{20} Расчет координаты случайного вектора.

$t_k = 0.5$, $x_k = \begin{pmatrix} 5.183 \\ 6.262 \end{pmatrix}$ и $y_j := x_k + t_k \cdot \xi_j, \quad y_j = \begin{pmatrix} 4.833 \\ 5.906 \end{pmatrix}$ - координаты случайной точки

Рис. 2.28. Листинг программы расчета минимума функции методом случайного поиска с возвратом при неудачном шаге (Пример 1.1, часть 28)

4^{20} Проверить выполнение условий: удачного шага и окончания расчета по M

Проверка уловия удачного шага

$f(x_k) = 0.203 \;\; f(y_j) = 0.121$ сравниваемые величины

проверка_шага := if(f(y_j) < f(x_k), шу, шн)

проверка_шага = "Шаг удачен. x_k+1=y_j, t_k+1=t_k, k=k+1. Усл. окончания по N"

Положить $x_{k+1} := y_j, \;\; x_{k+1} = \begin{pmatrix} 4.833 \\ 5.906 \end{pmatrix}$, $t_{k+1} := t_k$,

$k := k + 1 = 7$

Выполнение условия окончания расчета по N

$k = 7 \qquad N = 10$ контролируемые параметры

условие_оконч_по_N := if(k < N, k_меньше_N, k_равно_N)

условие_оконч_по_N = "Положить j=0. Перейти к шагу 2."

$j := 0$

Итерация $k = 7$ точек x_k

2^{21} Расчет случайного вектора ξ при $j = 0$ итерации ξ_j

$\mathrm{runif}(\mathrm{last}(x0) + 1, -1, 1) = \begin{pmatrix} 0.464 \\ -0.441 \end{pmatrix}$ - расчет случайного вектора

равномерно распеределенный в итервале [-1 1] $\zeta := \begin{pmatrix} 0.364 \\ 0.444 \end{pmatrix}$ - случайный вектор

Рис. 2.29. Листинг программы расчета минимума функции методом случайного поиска с возвратом при неудачном шаге 1 (Пример 1.1, часть 29)

$\xi_j := \frac{\zeta}{|\zeta|}$, $\xi_j = \begin{pmatrix} 0.634 \\ 0.773 \end{pmatrix}$ - случайный вектор единичной длины

$|\xi_j| = 1$, определяющий направление шага

3[21] Расчет координаты случайного вектора.

$t_k = 0.5$, $x_k = \begin{pmatrix} 4.833 \\ 5.906 \end{pmatrix}$ и $y_j := x_k + t_k \cdot \xi_j$, $y_j = \begin{pmatrix} 5.15 \\ 6.292 \end{pmatrix}$ - координаты случайной точки

4[21] Проверить выполнение условий: удачного шага и окончания расчета по M

Проверка уловия удачного шага

$f(x_k) = 0.121$ $f(y_j) = 0.175$ сравниваемые величины

проверка_шага := if(f(y_j) < f(x_k) ,шу ,шн)

проверка_шага = "Шаг не удачен. Оценить число неудач - M (Шаг 5)."

5[14] Оценить число локальных неудач по параметру M при расчете случайных точек

$j = 0$, $M = 3$ - контролируемые параметры

условие_оконч_по_M := if(j < M ,j_меньше_M ,j_равно_M)

условие_оконч_по_M = "Положить j=j+1. Перейти к шагу 2."

$j := j + 1 = 1$

2[22] Расчет случайного вектора ξ при $j = 1$ итерации ξ_j

$\mathrm{runif}(\mathrm{last}(x0) + 1, -1, 1) = \begin{pmatrix} 0.364 \\ 0.444 \end{pmatrix}$ - расчет случайного вектора равномерно распеределенный в итервале [-1 1] $\zeta := \begin{pmatrix} -0.754 \\ 0.669 \end{pmatrix}$ - случайный вектор

Рис. 2.30. Листинг программы расчета минимума функции методом случайного поиска с возвратом при неудачном шаге 1 (Пример 1.1, часть 30)

$\xi_j := \frac{\zeta}{|\zeta|}$, $\xi_j = \begin{pmatrix} -0.748 \\ 0.664 \end{pmatrix}$ - случайный вектор единичной длины $|\xi_j| = 1$, определяющий направление шага

3[22] Расчет координаты случайного вектора.

$t_k = 0.5$, $x_k = \begin{pmatrix} 4.833 \\ 5.906 \end{pmatrix}$ и $y_j := x_k + t_k \cdot \xi_j$, $y_j = \begin{pmatrix} 4.459 \\ 6.238 \end{pmatrix}$ - координаты случайной точки

4[22] Проверить выполнение условий: удачного шага и окончания расчета по М

<u>Проверка уловия удачного шага</u>

$f(x_k) = 0.121$ $f(y_j) = 1.228$ сравниваемые величины

проверка_шага := if(f(y_j) < f(x_k), шу, шн)

проверка_шага = "Шаг не удачен. Оценить число неудач - М (Шаг 5)."

5[14] Оценить число локальных неудач по параметру М при расчете случайных точек

$j = 1$, $M = 3$ - контролируемые параметры

условие_оконч_по_М := if(j < M, j_меньше_М, j_равно_М)

условие_оконч_по_М = "Положить j=j+1. Перейти к шагу 2."

$j := j + 1 = 2$

2[23] Расчет случайного вектора ξ при $j = 2$ итерации ξ_j

Рис. 2.31. Листинг программы расчета минимума функции методом случайного поиска с возвратом при неудачном шаге 1 (Пример 1.1, часть 31)

$\mathrm{runif}(\mathrm{last}(x0)+1,-1,1)=\begin{pmatrix}-0.754\\0.669\end{pmatrix}$ - расчет случайного вектора равномерно распередeленный в итервале [-1 1] $\zeta := \begin{pmatrix}0.034\\-0.148\end{pmatrix}$ - случайный вектор

$\xi_j := \frac{\zeta}{|\zeta|},\ \xi_j = \begin{pmatrix}0.224\\-0.975\end{pmatrix}$ - случайный вектор единичной длины $|\xi_j| = 1$, определяющий направление шага

3[23] Расчет координаты случайного вектора.

$t_k = 0.5$, $x_k = \begin{pmatrix}4.833\\5.906\end{pmatrix}$ и $y_j := x_k + t_k \cdot \xi_j,\ y_j = \begin{pmatrix}4.945\\5.418\end{pmatrix}$ - координаты случайной точки

4[23] Проверить выполнение условий: удачного шага и окончания расчета по М

Проверка уловия удачного шага

$f(x_k) = 0.121$ $f(y_j) = 0.35$ сравниваемые величины

проверка_шага := if(f(y_j) < f(x_k), шу, шн)

проверка_шага = "Шаг не удачен. Оценить число неудач - М (Шаг 5)."

5[15] Оценить число локальных неудач по параметру М при расчете случайных точек

$j = 2$, $M = 3$ - контролируемые параметры

условие_оконч_по_М := if(j < M, j_меньше_М, j_равно_М)

условие_оконч_по_М = "Положить j=j+1. Перейти к шагу 2." $j := j + 1 = 3$

Рис. 2.32. Листинг программы расчета минимума функции методом случайного поиска с возвратом при неудачном шаге 1 (Пример 1.1, часть 32)

2^{24} Расчет случайного вектора ξ при $j = 3$ итерации ξ_j

$\mathrm{runif}(\mathrm{last}(x0) + 1, -1, 1) = \begin{pmatrix} 0.034 \\ -0.148 \end{pmatrix}$ - расчет случайного вектора равномерно распеределенный в итервале [-1 1] $\zeta := \begin{pmatrix} 0.899 \\ 0.099 \end{pmatrix}$ - случайный вектор

$\xi_j := \frac{\zeta}{|\zeta|}$, $\xi_j = \begin{pmatrix} 0.994 \\ 0.109 \end{pmatrix}$ - случайный вектор единичной длины $|\xi_j| = 1$, определяющий направление шага

3^{24} Расчет координаты случайного вектора.

$t_k = 0.5$, $x_k = \begin{pmatrix} 4.833 \\ 5.906 \end{pmatrix}$ и $y_j := x_k + t_k \cdot \xi_j$, $y_j = \begin{pmatrix} 5.33 \\ 5.961 \end{pmatrix}$ - координаты случайной точки

4^{24} Проверить выполнение условий: удачного шага и окончания расчета по M

Проверка уловия удачного шага

$f(x_k) = 0.121$ $f(y_j) = 0.437$ сравниваемые величины

проверка_шага := if(f(y_j) < f(x_k), шу, шн)

проверка_шага = "Шаг не удачен. Оценить число неудач - M (Шаг 5)."

5^{16} Оценить число локальных неудач по параметру M при расчете случайных точек

$j = 3$, $M = 3$ - контролируемые параметры

Рис. 2.33. Листинг программы расчета минимума функции методом случайного поиска с возвратом при неудачном шаге 1 (Пример 1.1, часть 33)

условие_оконч_по_M := if(j < M, j_меньше_M, j_равно_M)

условие_оконч_по_M = "Проверить условие окончания по R"

Проверить окончание расчета по минимальной величине шага

условие_оконч_по_R := if$\left(t_k \le R, \text{tk_меньше_R}, \text{tk_больше_R}\right)$

условие_оконч_по_R = "Расчет завершен. Точка минимума x*=xk"

Результаты расчета

$XR_{min} := x_{last(x)}$ $XR_{min} = \begin{pmatrix} 4.833 \\ 5.906 \end{pmatrix}$ координата точки минимума

$fXR_{min} := f\left(XR_{min}\right) = 0.121$ значение функции в точке минимума

Xx := x вектор расчетных значений точек при положительном шаге

$$Xx^T = \left[\begin{pmatrix} 8 \\ 9 \end{pmatrix} \begin{pmatrix} 7.243 \\ 8.347 \end{pmatrix} \begin{pmatrix} 6.245 \\ 8.411 \end{pmatrix} \begin{pmatrix} 6.135 \\ 7.417 \end{pmatrix} \begin{pmatrix} 5.378 \\ 6.763 \end{pmatrix} \begin{pmatrix} 5.268 \\ 5.77 \end{pmatrix} \begin{pmatrix} 5.183 \\ 6.262 \end{pmatrix} \begin{pmatrix} 4.833 \\ 5.906 \end{pmatrix}\right]$$

****

РАСЧЕТ ЗАВЕРШЕН

****

==

Реализация алгоритма метода случайного поиска с возвратом при неудачном шаге с помощью пользовательской функции

Подключить файл "Функция СМ ВНШ.xmcd" с описаниями собственных функций

Ссылка:E:\Студенты\Optimiz\многомерная численная 0 порядка\Поиск с возвратом при неудачном шаге\Функция СМ ВНШ.xm

Рис. 2.34. Листинг программы расчета минимума функции методом случайного поиска с возвратом при неудачном шаге 1 (Пример 1.1, часть 34)

Исходные параметры

$x0 = \begin{pmatrix} 8 \\ 9 \end{pmatrix}$ $\quad M = 3 \quad N = 10 \quad t0 = 1 \quad R = 0.5 \quad \beta = 0.5$

$СП\_ВНШ := F\_СП\_ВНШ(f, x0, M, N, t0, R, \beta, вх)$

$СП\_ВНШ^T = (\{2,1\} \ \ \{11,1\} \ \ 19)$ результаты расчета (функция возвращает вектор из трех параметров: 1 элемент - коррдинаты точки минимума функции, 2 элемент - вектор расчетных значений точек при положительном шаге, 3 злемент - общее количество точек расчитанных при реализации метода (удачных и не удачных)

Результаты расчета

$XR1_{min} := СП\_ВНШ_0 = \begin{pmatrix} 4.851 \\ 7.137 \end{pmatrix}$ точка минимума

$fXR1_{min} := f\left(XR1_{min}\right) = 1.382$ значение функции в точке минимума

$кол\_точек := СП\_ВНШ_2 = 19$ общее количество точек расчитанных при реализации метода (удачных и не удачных)

$X1z := СП\_ВНШ_1$ Вектор расчетных значений точек

$X1z^T = \blacksquare$

::

элементы вектора X1z с 0 по 6

$X1z0\_9 := submatrix(X1z, 0, 6, 0, 0)$

$$X1z0\_9^T = \left[\begin{pmatrix} 8 \\ 9 \end{pmatrix} \begin{pmatrix} 7.511 \\ 8.128 \end{pmatrix} \begin{pmatrix} 6.749 \\ 8.775 \end{pmatrix} \begin{pmatrix} 6.833 \\ 7.779 \end{pmatrix} \begin{pmatrix} 6.62 \\ 7.326 \end{pmatrix} \begin{pmatrix} 6.349 \\ 7.747 \end{pmatrix} \begin{pmatrix} 5.967 \\ 8.068 \end{pmatrix} \right]$$

Рис. 2.35. Листинг программы расчета минимума функции методом случайного поиска с возвратом при неудачном шаге 1 (Пример 1.1, часть 35)

элементы вектора X1z с 7 по конца

X1z7_end := submatrix(X1z ,7 ,last(X1z) ,0 ,0)

$$X1z7\_end^T = \left[\begin{pmatrix} 5.715 \\ 7.636 \end{pmatrix} \begin{pmatrix} 5.266 \\ 7.417 \end{pmatrix} \begin{pmatrix} 4.851 \\ 7.137 \end{pmatrix} \text{"Выход по R. Есть минимум."}\right]$$

::

****

Аналитический расчет точки минимума функции

$f1(x,y) := 4(x-5)^2 + (y-6)^2$ исследуемая функция (формальные параметры записаны в виде двух переменных)

$xx := 8$ $yy := 9$ начальное приближение координаты минимума

Given

$\frac{d}{dxx} f1(xx,yy) = 0$ $\frac{d}{dyy} f1(xx,yy) = 0$

$XT_{min} := Minerr(xx,yy)$ $XT_{min} = \begin{pmatrix} 5 \\ 6 \end{pmatrix}$ координата точки минимума

$fXT_{min} := f1\left(XT_{min_0}, XT_{min_1}\right) = 0$ значение функции в точке минимума

Погрешность численного расчета

$\Delta X\% := \frac{\left|XR_{min} - XT_{min}\right|}{\left|XR_{min}\right|} \cdot 100 = 2.515$ погрешность определения координат точки минимума

Рис. 2.36. Листинг программы расчета минимума функции методом случайного поиска с возвратом при неудачном шаге 1 (Пример 1.1, часть 36)

$$\Delta f\% := \frac{fXR1_{min} - fXT_{min}}{fXR1_{min}} \cdot 100 = 100$$ погрешность определения значения функции в точке минимума

<u>Вывод</u>: Погрешность определения координаты точки минимума
$\Delta X\% = 2.515$
процента, погрешность определения значения функции в точке минимума $\Delta f\% = 100$ процентов.

Графическое отображение результатов расчета

Построить трехмерный график в виде контурных линий в диапазоне значений -3<x<8, -3<y<9, с шагом 0.1, нанести расчетные точки.

<u>Расчет матрицы для постоения поверхности функции, используя дискретный аргумент.</u>

$x_{min} := 0 \quad x_{max} := 10 \quad y_{min} := -3 \quad y_{max} := 14 \quad \Delta x := 0.1 \quad \Delta y := 0.1$

кол. точек расчета

$$Ni := \frac{x_{max} - x_{min}}{\Delta x} \quad Ni = 100 \quad Nj := \frac{y_{max} - y_{min}}{\Delta y} \quad Nj = 170$$

$i := 0..Ni \qquad j := 0..Nj$

$x1_{i,j} := x_{min} + \Delta x \cdot i \qquad y1_{i,j} := y_{min} + \Delta y \cdot j \qquad FM := \overrightarrow{f1(x1, y1)}$

<u>Теоретические координаты точки минимума (красная точка)</u>

$X12_0 := XT_{min_0} = 5$ значение x-составляющей

$Y12_0 := XT_{min_1} = 6$ значение y-составляющей

$Z12_0 := fXT_{min} = 0$ значение z-составляющей

Рис. 2.37. Листинг программы расчета минимума функции методом случайного поиска с возвратом при неудачном шаге 1 (Пример 1.1, часть 37)

Расчетные координаты точки минимума (черный крест)

$XT12_0 := XR1_{min_0} = 4.851$ значение x-составляющей

$YT12_0 := XR1_{min_1} = 7.137$ значение y-составляющей

$ZT12_0 := fXR1_{min} = 1.382$ значение z-составляющей

Координаты всех расчетных точек (черная кривая)

Преобразование расчетных точек в двухстолбцовый массив и расчет значений функции в этих точках. (Преобразование необходимы для построения точек на графике)

$X2z := submatrix(X1z, 0, last(X1z) - 1, 0, 0)$ удаление из вектора последнего элемента

$X2G := v\_k(X2z)$ преобразование блочного вектора, каждый элемент которого - вектор из двух элементов, в матрицу из двух столбцов

$fX2G := \overrightarrow{f1\left(X2G^{\langle 0\rangle}, X2G^{\langle 1\rangle}\right)}$ значения функции в расчетных точках.

$fX2G^T = (45 \quad 29.753 \quad 19.943 \quad 16.61 \quad 12.263 \quad 10.335 \quad 8.016 \quad 4.722 \quad 2.289 \quad 1.382)$

Рис. 2.38. Листинг программы расчета минимума функции методом случайного поиска с возвратом при неудачном шаге 1 (Пример 1.1, часть 38)

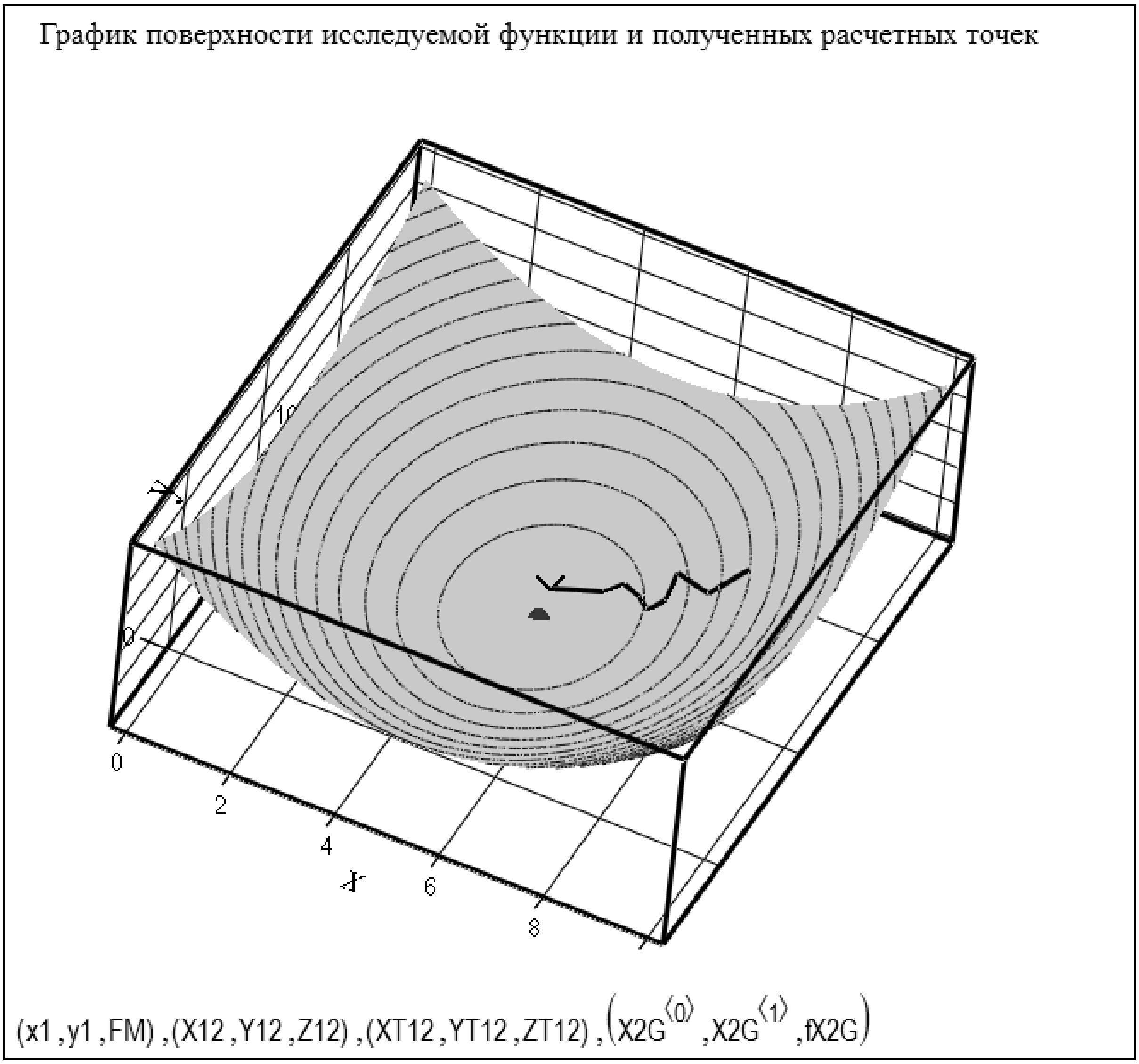
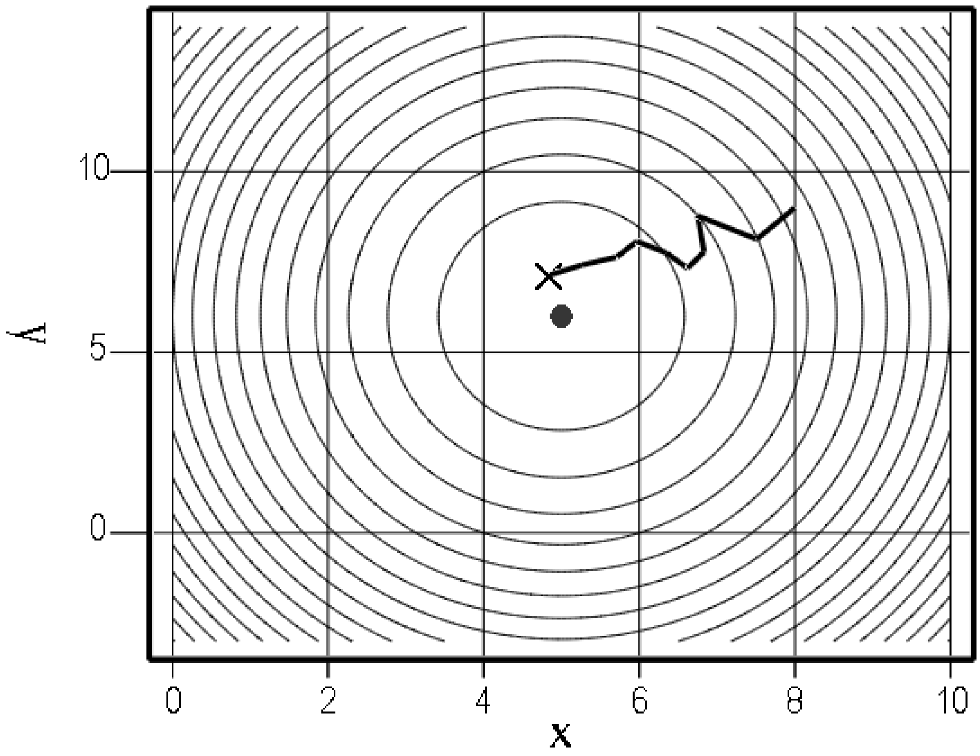


Рис. 2.39. Листинг программы расчета минимума функции методом случайного поиска с возвратом при неудачном шаге 1 (Пример 1.1, часть 39)

График линий уровня исследуемой функции и полученных расчетных точек



(x1 ,y1 ,FM) ,(X12 ,Y12 ,Z12) ,(XT12 ,YT12 ,ZT12) ,$\left(X2G^{\langle 0\rangle}, X2G^{\langle 1\rangle}, fX2G\right)$

Рис. 2.40. Листинг программы расчета минимума функции методом случайного поиска с возвратом при неудачном шаге 1 (Пример 1.1, часть 40)

2.2 Листинги с описанием собственных функций для поиска минимума функции методом случайного поиска с возвратом при неудачном шаге в пакете MathCAD 15

Функция, реализующая случайный метод с возвратом при неудачном шаге (СМ ВНШ)

(Описание функции)

На входе: fz - имя исследуемой функции, x0z - вектор координат исходной точки, Mz - максимальный индекс числа неудачно выполненных испытаний на текущей итерации, Nz - предельный индекс числа итераций при удачном шаге, t0z - начальный шаг, Rz - минимальный шаг, βz - кооэфициент расширения, вх - вектор размером 2x1 - информационные сообщения о причинах завершения расчетов.
На выходе: блочный вектор размером 3x1: 1 элемент - vx_{min}- расчетная точка минимума, 2 элемент - vx - координаты всех удачных расчетных точек в виде блочного вектора, 3 элемент - jj - общее количество рачетов (удачных и не удачных).

$$вх := \begin{pmatrix} \text{"Выход по N. Есть минимум."} \\ \text{"Выход по R. Есть минимум."} \end{pmatrix}$$

```
F_СП_ВНШ(fz,x0z,Mz,Nz,t0z,Rz,βz,вх) := | jj ← 0
                                       | k ← 0
                                       | j ← 0
                                       | vx_k ← x0z
                                       | vt_k ← t0z
                                       | while  k < Nz
                                       |    | vξ_j ← runif(last(x0z) + 1,−1,1)
```

Рис. 2.41. Листинг с описанием собственной функции, реализующей поиск минимума функции методом случайного поиска с возвратом при неудачном шаге (часть 1)

```
vy_j ← vx_k + vt_k · vξ_j / |vξ_j|
jj ← jj + 1
if  fz(vy_j) < fz(vx_k)
    vx_{k+1} ← vy_j
    vt_{k+1} ← vt_k
    k ← k + 1
    j ← 0  if  k < Nz
    otherwise
        vx_min ← vx_k
        vx_{k+1} ← вх_0
otherwise
    j ← j + 1  if  j < Mz
    otherwise
        if  vt_k ≤ Rz
            vx_min ← vx_k
            vx_{k+1} ← вх_1
            return (vx_min, vx, jj)
```

Рис. 2.42. Листинг с описанием собственной функции, реализующей поиск минимума функции методом случайного поиска с возвратом при неудачном шаге (часть 2)

```
otherwise
  vt_k ← βz·vt_k
  j ← 0
(vx_min)
(vx    )
(jj    )
```

Рис. 2.43. Листинг с описанием собственной функции, реализующей поиск минимума функции методом случайного поиска с возвратом при неудачном шаге (часть 3)

2.3 Листинги программы поиска минимума функции Хаммельблау методом случайного поиска с возвратом при неудачном шаге в пакете MathCAD 15 (Пример 1.2)

Пример №1.2: Найти локальный минимум функции Хаммельблау методом случайного поиска с возвратом при не удачном шаге
Последовательность вычислений
Численный метод: реализовать алгоритм :
1. Построчно. Построить последовательность расчетных точек и рассчитать значение функции в этих точках, проверить выполнение условий, т.о. определить точку минимума.
2. С использованием одной пользовательской функции.
3.Сравнить результаты расчета
График. построить трехмерный график, нанести последовательность расчетных точек и точку локального минимума
Построить два типа графиков: 1. в виде поверхности. 2. в виде линий уровня

Вид функции Хаммельблау напоминает глубокую яму с четырьмя углублениями и одной выпуклостью на дне, что сильно осложняет реализацию многих алгоритмов оптимизации

Вспомогательные функции и переменные, используемые при построчной реализации алгоритма
1. Функция для пересчета из блочного вектора в массив из двух колонок. На входе:xr- блочный вектор точек расчета.
На выходе: массив из двух колонок (x и y) - координат расчетных точек. Используется для подготовки данных расчета при построении графиков

$$v\_k(xr) := \left| \begin{array}{l} vx \leftarrow {xr_0}^T \\ \text{for } \ i \in 1..\text{last}(xr) \\ \quad vx \leftarrow \text{stack}\left(vx, {xr_i}^T\right) \\ vx \end{array} \right.$$

Рис. 2.44. Листинг программы расчета минимума функции Хаммельблау методом случайного поиска с возвратом при неудачном шаге
(Пример 1.2, часть 1)

2. Переменные (шу и шн), используются для оценки успешности шага по случайному направлению с помощью оператора -

проверка_шага := $\mathrm{if}\left(f\left(y_j\right) < f\left(x_k\right), \text{шу}, \text{шн}\right)$

шу := "Шаг удачен. x_k+1=y_j, t_k+1=t_k, k=k+1. Усл. окончания по N"

шн := "Шаг не удачен. Оценить число неудач - M (Шаг 5)."

3. Переменные (оконч_k_меньше_N и оконч_k_равно_N), используются для оценки успешности
окончания расчета по параметру N с помощью оператора -

условие_оконч_по_N := if(k < N, k_меньше_N, k_равно_N)

k_меньше_N := "Положить j=0. Перейти к шагу 2."

k_равно_N := "k=N. Расчет завершен. Точка минимума x*=xk"

4. Переменные (оконч_j_меньше_M и оконч_j_равно_M), используются для оценки успешности
окончания расчета по параметру M с помощью оператора -

условие_оконч_по_M := if(j < M, j_меньше_M, j_равно_M)

j_меньше_M := "Положить j=j+1. Перейти к шагу 2."

j_равно_M := "Проверить условие окончания по R"

5. Переменные (tk_меньше_R и tk_больше_R), используются для оценки успешности
окончания расчета по параметру R с помощью оператора -

условие_оконч_по_R := $\mathrm{if}\left(t_k \le R, \text{tk\_меньше\_R}, \text{tk\_больше\_R}\right)$

tk_меньше_R := "Расчет завершен. Точка минимума x*=xk"

tk_больше_R := "Положить tk =β*tk, j=0, перейти к шагу 2"

Построчная реализация алгоритма метода случайного поиска с возвратом при неудачном шаге

Рис. 2.45. Листинг программы расчета минимума функции Хаммельблау методом случайного поиска с возвратом при неудачном шаге
(Пример 1.2, часть 2)

1^0. Исходные данные и описание переменных, использованных при построчной реализации алгоритма

$f(x) := \left[(x_0)^2 + x_1 - 11\right]^2 + \left[x_0 + (x_1)^2 - 7\right]^2$ исследуемая функция Хаммельблау

$x0 := \begin{pmatrix} 0 \\ -1 \end{pmatrix}$ - начальная точка

$\beta := 0.5$ - коэффициент сжатия $0<\beta<1$

$M := 3$ - максимальный индекс числа неудачно выполненных испытаний на текущей итерации (всего 4 неудачи, так как для первого элемента $j := 0$)

$N := 10$ - предельный индекс числа итераций при удачном шаге (всего 11 удачных шагов, так как для первого элемента $k := 0$)

$t0 := 1$ - начальная величина шага

$R := 0.5$ - минимальная величина шага

$k := 0$ - индекс итерации переменной (точки) x_k при удачном шаге

x - вектор с координатами точек при удачном шаге

ξ - случайный вектор единичной длины, определяющий направление шага

$j := 0$ - индекс итерации случайной переменной (точки) - y_j

y - вектор с координатами случайных точек при случайном выборе направления шага

Внимание: В учебных целях значение для случайного вектора рассчитывается с помощью функции $\mathrm{runif}(2,-1,1)$ на каждом шаге, а затем копируется в переменную ζ и значение для ξ становится постоянным для данного примера. В противном случае не удается добиться стабильно работающего алгоритма для этого метода. При написании пользовательской функции, реализующий алгоритм, этого допущения нет.

Рис. 2.46. Листинг программы расчета минимума функции Хаммельблау методом случайного поиска с возвратом при неудачном шаге (Пример 1.2, часть 3)

****

РАСЧЕТ ЗАВЕРШЕН

****

===

Реализация алгоритма метода случайного поиска с возвратом при неудачном шаге с помощью пользовательской функции

Подключить файл "Функция СМ ВНШ.xmcd" с описаниями собственных функц

Ссылка:G:\Пособие по оптимизации Володя 2020\Поиск с возвратом при неудачном шаге пособие к Володе 2019-2020

Исходные параметры

$x0 = \begin{pmatrix} 0 \\ -1 \end{pmatrix}$ $M = 3$ $N = 10$ $t0 = 1$ $R = 0.5$ $\beta = 0.5$

$СП\_ВНШ := F\_СП\_ВНШ(f, x0, M, N, t0, R, \beta, вх)$

$СП\_ВНШ^T = (\{2,1\} \ \ \{9,1\} \ \ 20)$ результаты расчета (функция возвращает вектор из трех парамертов: 1 элемент - коррдинаты точки минимума функции, 2 элемент - вектор расчетных значений точек при положительном шаге, 3 злемент - общее количество точек расчитанных при реализации метода (удачных и не удачных)

Результаты расчета

$XR1_{min} := СП\_ВНШ_0 = \begin{pmatrix} 3.489 \\ -1.006 \end{pmatrix}$ точка минимума

$fXR1_{min} := f(XR1_{min}) = 6.278$ значение функции в точке минимума

$кол\_точек := СП\_ВНШ_2 = 20$ общее количество точек расчитанных при реализации метода (удачных и не удачных)

Рис. 2.47. Листинг программы расчета минимума функции Хаммельблау методом случайного поиска с возвратом при неудачном шаге
(Пример 1.2, часть 4)

$X1z := СП\_ВНШ_1$ Вектор расчетных значений точек

$$X1z^T = \left[\begin{pmatrix}0\\-1\end{pmatrix}\begin{pmatrix}0.618\\-1.786\end{pmatrix}\begin{pmatrix}1.618\\-1.783\end{pmatrix}\begin{pmatrix}1.709\\-0.787\end{pmatrix}\begin{pmatrix}2.128\\0.121\end{pmatrix}\begin{pmatrix}3.086\\0.408\end{pmatrix}\begin{pmatrix}3.222\\-0.583\end{pmatrix}\begin{pmatrix}3.489\\-1.006\end{pmatrix}\right.$$

::

элементы вектора X1z с 0 по 6

$X1z0\_6 := \mathrm{submatrix}(X1z, 0, 5, 0, 0)$

$$X1z0\_6^T = \left[\begin{pmatrix}0\\-1\end{pmatrix}\begin{pmatrix}0.618\\-1.786\end{pmatrix}\begin{pmatrix}1.618\\-1.783\end{pmatrix}\begin{pmatrix}1.709\\-0.787\end{pmatrix}\begin{pmatrix}2.128\\0.121\end{pmatrix}\begin{pmatrix}3.086\\0.408\end{pmatrix}\right]$$

элементы вектора X1z с 7 по конца

$X1z7\_end := \mathrm{submatrix}(X1z, 6, \mathrm{last}(X1z), 0, 0)$

$$X1z7\_end^T = \left[\begin{pmatrix}3.222\\-0.583\end{pmatrix}\begin{pmatrix}3.489\\-1.006\end{pmatrix} \text{"Выход по R. Есть минимум."}\right]$$

Параметр по которому происходит завершение расчета $X1_v := X1z_{\mathrm{last}(X1z)} =$ "Выход по R. Есть минимум."

::

****

Вывод: Точка минимума равна $XR1_{min} = \begin{pmatrix}3.489\\-1.006\end{pmatrix}$, значение функции в точке минимума равно $f(XR1_{min}) = 6.278$

****

Рис. 2.48. Листинг программы расчета минимума функции Хаммельблау методом случайного поиска с возвратом при неудачном шаге (Пример 1.2, часть 5)

Аналитический расчет точки минимума функции

$f\_X(x,y) := \left(x^2 + y - 11\right)^2 + \left(x + y^2 - 7\right)^2$ исследуемая функция (формальные параметры записаны в виде двух переменных)

1. Получить аналитические выражения производных фукции по переменным x,y, используя символьный метод

$$\frac{d}{dx} f\_X(x,y) \rightarrow 2\cdot x + 4\cdot x\cdot\left(x^2 + y - 11\right) + 2\cdot y^2 - 14$$

$$\frac{d}{dy} f\_X(x,y) \rightarrow 2\cdot y + 4\cdot y\cdot\left(y^2 + x - 7\right) + 2\cdot x^2 - 22$$

2. Найти минимумы функций, используя блок расчета Given-MinErr. В качестве уравнений использовать выражения для первых производных (скопировав их).
При решении использовать метод LevenbergMarquardt.

$xx := \begin{pmatrix} -0.5 \\ -3 \\ 3 \\ 3 \end{pmatrix}$ $yy := \begin{pmatrix} -1 \\ 3 \\ -3 \\ 3 \end{pmatrix}$ начальные приближение

Given

$$4\cdot\left(xx^2 + yy - 11\right)\cdot xx + 2\cdot xx + 2\cdot yy^2 - 14 = 0$$

$$2\cdot xx^2 + 2\cdot yy - 22 + 4\cdot\left(xx + yy^2 - 7\right)\cdot yy = 0$$

координата точки минимума

$XT_{min} := \overrightarrow{Minerr(xx,yy)}$ $XT_{min}{}^T = \left[\begin{pmatrix} -3.779 \\ -2.805 \\ 3.584 \\ 3 \end{pmatrix} \begin{pmatrix} -3.283 \\ 3.131 \\ -1.848 \\ 2 \end{pmatrix}\right]$

Рис. 2.49. Листинг программы расчета минимума функции Хаммельблау методом случайного поиска с возвратом при неудачном шаге
(Пример 1.2, часть 6)

$$fXT_{min} := f\_X\left(XT_{min_0}, XT_{min_1}\right) = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$ значение функции в точке минимума

3. Определить максимум функции на "дне ямы" используя блок given-Maximize

$xx0 := 0$ $yy0 := 0$ начальное приближение координат точки максимума

Given

$xx0 \geq -3.5$ $yy0 \geq -3.5$

$xx0 \leq 3.5$ $yy0 \leq 3.5$

Ограничения в виде нераветств. Ограничивается зона дна ямы. В противном случае максимум всегда будет на стенке ямы.

$X_{max} := \mathrm{Maximize}(f\_X, xx0, yy0)$ $X_{max} = \begin{pmatrix} -0.271 \\ -0.923 \end{pmatrix}$ Координаты точки максимума на дне ямы

$fX_{max} := f\_X\left(X_{max_0}, X_{max_1}\right) = 181.617$ Значение функции в точке максимума

Графическое отображение результатов расчета

Построить трехмерный график в виде контурных линий в диапазоне значений -6<x<6, -6<y<6, с шагом 0.1, нанести расчетные точки.

Расчет матрицы для постоения поверхности функции, используя дискретный аргумент.

$x_{min} := -6$ $x_{max} := 6$ $\Delta x := 0.1$

$y_{min} := -6$ $y_{max} := 6$ $\Delta y := 0.1$

Рис. 2.50. Листинг программы расчета минимума функции Хаммельблау методом случайного поиска с возвратом при неудачном шаге (Пример 1.2, часть 7)

$$Ni := \frac{x_{max} - x_{min}}{\Delta x} \qquad Ni = 120$$

кол. точек расчета

$$Nj := \frac{y_{max} - y_{min}}{\Delta y} \qquad Nj = 120$$

$$i := 0..Ni \qquad j := 0..Nj$$

$$x1_{i,j} := x_{min} + \Delta x \cdot i \qquad y1_{i,j} := y_{min} + \Delta y \cdot j \qquad FM := \overrightarrow{f\_X(x1, y1)}$$

<u>Теоретические координаты точек минимума (красная точка)</u>

значение x-составляющей

$$X12 := XT_{min_0} \qquad X12^T = (-3.779 \quad -2.805 \quad 3.584 \quad 3)$$

значение y-составляющей

$$Y12 := XT_{min_1} \qquad Y12^T = (-3.283 \quad 3.131 \quad -1.848 \quad 2)$$

значение z-составляющей

$$Z12 := fXT_{min} \qquad Z12^T = (0 \quad 0 \quad 0 \quad 0)$$

<u>Рассчитанные по алгоритму координаты точки минимума (черный крест)</u>

$XT12_0 := XR1_{min_0} = 3.489$ значение x-составляющей

$YT12_0 := XR1_{min_1} = -1.006$ значение y-составляющей

$ZT12_0 := fXR1_{min} = 6.278$ значение z-составляющей

Рис. 2.51. Листинг программы расчета минимума функции Хаммельблау методом случайного поиска с возвратом при неудачном шаге
(Пример 1.2, часть 8)

<u>Координаты всех расчетных по алгоритму (черная кривая)</u>

Преобразование расчетных точек в двухстолбцовый массив и расчет значений функции в этих точках. (Преобразование необходимы для построения точек на графике)

$X2z := \mathrm{submatrix}(X1z, 0, \mathrm{last}(X1z) - 1, 0, 0)$ удаление из вектора последнего элемента

элементы вектора X2z с 0 по 6

$X2z0\_6 := \mathrm{submatrix}(X2z, 0, 5, 0, 0)$

$$X2z0\_6^T = \left[\begin{pmatrix} 0 \\ -1 \end{pmatrix} \begin{pmatrix} 0.618 \\ -1.786 \end{pmatrix} \begin{pmatrix} 1.618 \\ -1.783 \end{pmatrix} \begin{pmatrix} 1.709 \\ -0.787 \end{pmatrix} \begin{pmatrix} 2.128 \\ 0.121 \end{pmatrix} \begin{pmatrix} 3.086 \\ 0.408 \end{pmatrix}\right]$$

элементы вектора X2z с 7 до конца

$X2z7\_end := \mathrm{submatrix}(X2z, 6, \mathrm{last}(X2z), 0, 0)$

$$X2z7\_end^T = \left[\begin{pmatrix} 3.222 \\ -0.583 \end{pmatrix} \begin{pmatrix} 3.489 \\ -1.006 \end{pmatrix}\right]$$

$X2G := v\_k(X2z)$ преобразование блочного вектора в матрицу из двух столбцов

элементы вектора X2G с 0 по 6

$X2G0\_6 := \mathrm{submatrix}(X2G, 0, 5, 0, 1)$

$$X2G0\_6^T = \begin{pmatrix} 0 & 0.618 & 1.618 & 1.709 & 2.128 & 3.086 \\ -1 & -1.786 & -1.783 & -0.787 & 0.121 & 0.408 \end{pmatrix}$$

Рис. 2.52. Листинг программы расчета минимума функции Хаммельблау методом случайного поиска с возвратом при неудачном шаге
(Пример 1.2, часть 9)

элементы вектора X2G с 7 до конца

$$X2G7\_end := \mathrm{submatrix}\left(X2G, 6, \mathrm{last}\left(X2G^{\langle 0 \rangle}\right), 0, 1\right)$$

$$X2G7\_end^{T} = \begin{pmatrix} 3.222 & 3.489 \\ -0.583 & -1.006 \end{pmatrix}$$

::

$$fX2G := \overrightarrow{f\_X\left(X2G^{\langle 0 \rangle}, X2G^{\langle 1 \rangle}\right)}$$ значения функции в расчетных точках.

::

элементы вектора fX2G с 0 по 6

$$fX2G0\_6 := \mathrm{submatrix}(fX2G, 0, 5, 0, 0)$$

$$fX2G0\_6^{T} = \begin{pmatrix} 180 & 164.038 & 108.158 & 100.418 & 63.948 & 15.196 \end{pmatrix}$$

элементы вектора fX2G с 7 до конца

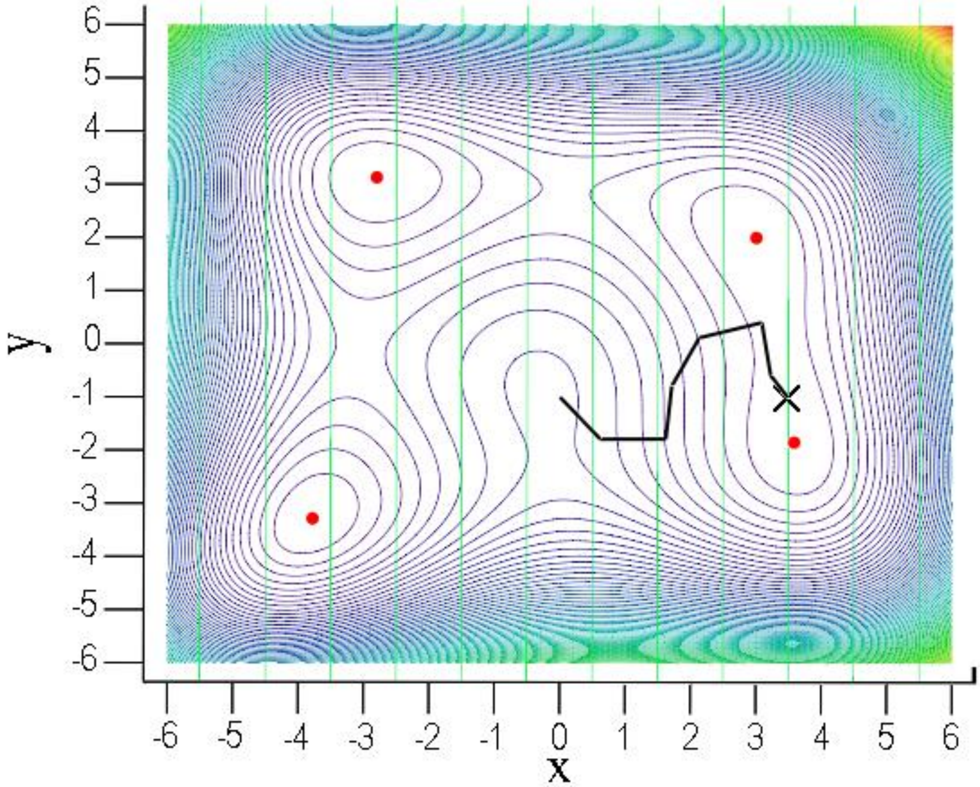
$$fX2G7\_end := \mathrm{submatrix}(fX2G, 6, \mathrm{last}(fX2G), 0, 0)$$

$$fX2G7\_end^{T} = \begin{pmatrix} 13.266 & 6.278 \end{pmatrix}$$

::

Рис. 2.53. Листинг программы расчета минимума функции Хаммельблау методом случайного поиска с возвратом при неудачном шаге
(Пример 1.2, часть 10)

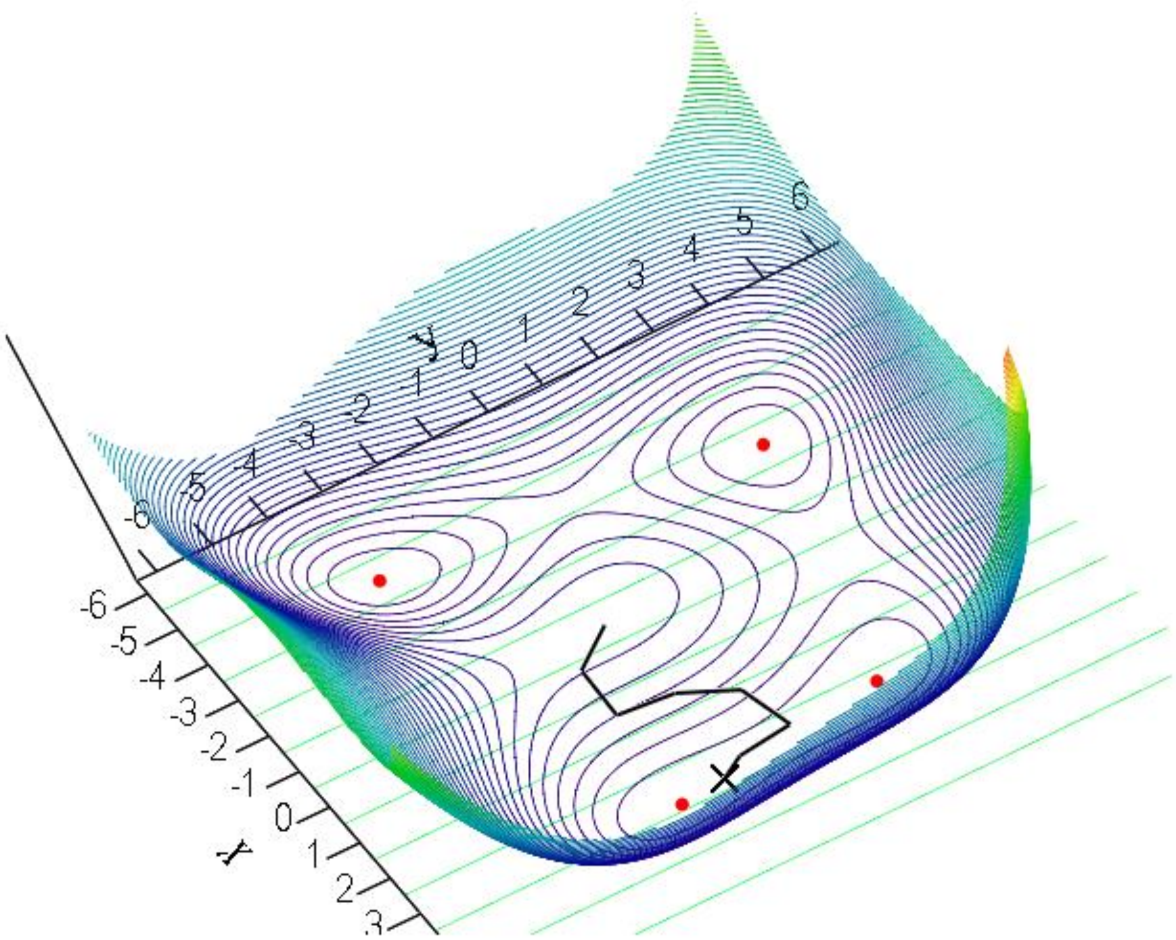
График в виде линий уровня исследуемой функции, теоретические и расчетные точки



$(x1, y1, FM), (X12, Y12, Z12), \left(X2G^{\langle 0\rangle}, X2G^{\langle 1\rangle}, fX2G\right), (XT12, YT12, ZT12)$

Рис. 2.54. Листинг программы расчета минимума функции Хаммельблау методом случайного поиска с возвратом при неудачном шаге (Пример 1.2, часть 11)

График в виде поверхности исследуемой функции, теретические и расчетные точки



$(x1, y1, FM), (X12, Y12, Z12), \left(X2G^{\langle 0 \rangle}, X2G^{\langle 1 \rangle}, fX2G\right), (XT12, YT12, ZT12)$

Рис. 2.55. Листинг программы расчета минимума функции Хаммельблау методом случайного поиска с возвратом при неудачном шаге
(Пример 1.2, часть 12)

2.4 Листинги программы поиска минимума функции Розенброка методом случайного поиска с возвратом при неудачном шаге в пакете MathCAD 15 (Пример 1.3)

Пример №1.3: Найти локальный минимум функции Розенброк методом случайного поиска с возвратом при не удачном шаге
Последовательность вычислений
Численный метод: реализовать алгоритм :
1. Построчно. Построить последовательность расчетных точек и рассчитать значение функции в этих точках, проверить выполнение условий, т.о. определить точку минимума.
2. С использованием одной пользовательской функции.
3.Сравнить результаты расчета
График. построить трехмерный график, нанести последовательность расчетных точек и точку локального минимума
Построить два типа графиков: 1. в виде поверхности. 2. в виде линий уровня

Вид функции Розенброка напоминает глубокий овраг с ямкой на дне, что сильно осложняет реализацию многих алгоритмов оптимизации.

Вспомогательные функции и переменные, используемые при построчной реализации алгоритма
1. Функция для пересчета из блочного вектора в массив из двух колонок. На входе: xr- блочный вектор точек расчета.
На выходе: массив из двух колонок (x и y) - координат расчетных точек. Используется для подготовки данных расчета при построении графиков

```
v_k(xr) := | vx ← xr_0^T
           | for  i ∈ 1..last(xr)
           |    vx ← stack(vx, xr_i^T)
           | vx
```

Рис. 2.56. Листинг программы расчета минимума функции Розенброка методом случайного поиска с возвратом при неудачном шаге
(Пример 1.3, часть 1)

2. Переменные (шу и шн), используются для оценки успешности шага по случайному направлению с помощью оператора -

проверка_шага := if$\left(f\left(y_j\right) < f\left(x_k\right), шу, шн\right)$

шу := "Шаг удачен. x_k+1=y_j, t_k+1=t_k, k=k+1. Усл. окончания по N"

шн := "Шаг не удачен. Оценить число неудач - M (Шаг 5)."

3. Переменные (оконч_k_меньше_N и оконч_k_равно_N), используются для оценки успешности
окончания расчета по параметру N с помощью оператора -

условие_оконч_по_N := if(k < N, k_меньше_N, k_равно_N)

k_меньше_N := "Положить j=0. Перейти к шагу 2."

k_равно_N := "k=N. Расчет завершен. Точка минимума x*=xk"

4. Переменные (оконч_j_меньше_M и оконч_j_равно_M), используются для оценки успешности
окончания расчета по параметру M с помощью оператора -

условие_оконч_по_M := if(j < M, j_меньше_M, j_равно_M)

j_меньше_M := "Положить j=j+1. Перейти к шагу 2."

j_равно_M := "Проверить условие окончания по R"

5. Переменные (tk_меньше_R и tk_больше_R), используются для оценки успешности
окончания расчета по параметру R с помощью оператора -

условие_оконч_по_R := if$\left(t_k \le R, tk_меньше_R, tk_больше_R\right)$

tk_меньше_R := "Расчет завершен. Точка минимума x*=xk"

tk_больше_R := "Положить tk =β*tk, j=0, перейти к шагу 2"

Построчная реализация алгоритма метода случайного поиска с возвратом при неудачном шаге

Рис. 2.57. Листинг программы расчета минимума функции Розенброка методом случайного поиска с возвратом при неудачном шаге
(Пример 1.3, часть 2)

1^0. Исходные данные и описание переменных, использованных при построчной реализации алгоритма

$f(x) := 100 \cdot \left[x_1 - (x_0)^2\right]^2 + (1 - x_0)^2$ исследуемая функция Розенброка

$x0 := \begin{pmatrix} 2 \\ 2 \end{pmatrix}$ - начальная точка

$\beta := 0.5$ - коэффициент сжатия $0<\beta<1$

$M := 3$ - максимальный индекс числа неудачно выполненных испытаний на текущей итерации (всего 4 неудачи, так как для первого элемента $j := 0$)

$N := 10$ - предельный индекс числа итераций при удачном шаге (всего 11 удачных шагов, так как для первого элемента $k := 0$)

$t0 := 1$ - начальная величина шага

$R := 0.0625$ - минимальная величина шага

$k := 0$ - индекс итерации переменной (точки) x_k при удачном шаге

x - вектор с координатами точек при удачном шаге

ξ - случайный вектор единичной длины, определяющий направление шага

$j := 0$ - индекс итерации случайной переменной (точки) - y_j

y - вектор с координатами случайных точек при случайном выборе направления шага

Внимание: В учебных целях значение для случайного вектора рассчитывается с помощью функции $runif(2,-1,1)$ на каждом шаге, а затем копируется в переменную ζ и значение для ξ становится постоянным для данного примера. В противном случае не удается добиться стабильно работающего алгоритма для этого метода. При написании пользовательской функции, реализующий алгоритм, этого допущения нет.

Рис. 2.58. Листинг программы расчета минимума функции Розенброка методом случайного поиска с возвратом при неудачном шаге
(Пример 1.3, часть 3)

****

РАСЧЕТ ЗАВЕРШЕН

****

==

Реализация алгоритма метода случайного поиска с возвратом при неудачном шаге с помощью пользовательской функции

Подключить файл "Функция СМ ВНШ.xmcd" с описаниями собственных функций

Ссылка:G:\Пособие по оптимизации Володя 2020\Поиск с возвратом при неудачном шаге пособие к Володе 2019-2

Исходные параметры

$x0 = \begin{pmatrix} 2 \\ 2 \end{pmatrix}$ $M = 3$ $N = 10$ $t0 = 1$ $R = 0.063$ $\beta = 0.5$

$СП\_ВНШ := F\_СП\_ВНШ(f, x0, M, N, t0, R, \beta, вх)$

$СП\_ВНШ^T = (\{2,1\}\ \ \{10,1\}\ \ 41)$ результаты расчета (функция возвращает вектор из трех параметров: 1 элемент - коррдинаты точки минимума функции, 2 элемент - вектор расчетных значений точек при положительном шаге, 3 злемент - общее количество точек расчитанных при реализации метода (удачных и не удачных)

Результаты расчета

$XR1_{min} := СП\_ВНШ_0 = \begin{pmatrix} 1.25 \\ 1.558 \end{pmatrix}$ точка минимума

$fXR1_{min} := f(XR1_{min}) = 0.064$ значение функции в точке минимума

$кол\_точек := СП\_ВНШ_2 = 41$ общее количество точек расчитанных при реализации метода (удачных и не удачных)

Рис. 2.59. Листинг программы расчета минимума функции Розенброка методом случайного поиска с возвратом при неудачном шаге (Пример 1.3, часть 4)

$X1z := СП\_ВНШ_1$ Вектор расчетных значений точек

$$X1z^T = \left[\begin{pmatrix} 2 \\ 2 \end{pmatrix} \begin{pmatrix} 1.05 \\ 1.689 \end{pmatrix} \begin{pmatrix} 1.497 \\ 1.912 \end{pmatrix} \begin{pmatrix} 1.275 \\ 1.465 \end{pmatrix} \begin{pmatrix} 1.207 \\ 1.569 \end{pmatrix} \begin{pmatrix} 1.315 \\ 1.631 \end{pmatrix} \begin{pmatrix} 1.218 \\ 1.552 \end{pmatrix} \begin{pmatrix} 1.244 \\ 1.495 \end{pmatrix} \right.$$

::

элементы вектора X1z с 0 по 6

$X1z0\_6 := \mathrm{submatrix}(X1z, 0, 5, 0, 0)$

$$X1z0\_6^T = \left[\begin{pmatrix} 2 \\ 2 \end{pmatrix} \begin{pmatrix} 1.05 \\ 1.689 \end{pmatrix} \begin{pmatrix} 1.497 \\ 1.912 \end{pmatrix} \begin{pmatrix} 1.275 \\ 1.465 \end{pmatrix} \begin{pmatrix} 1.207 \\ 1.569 \end{pmatrix} \begin{pmatrix} 1.315 \\ 1.631 \end{pmatrix} \right]$$

элементы вектора X1z с 7 по конца

$X1z7\_end := \mathrm{submatrix}(X1z, 6, \mathrm{last}(X1z), 0, 0)$

$$X1z7\_end^T = \left[\begin{pmatrix} 1.218 \\ 1.552 \end{pmatrix} \begin{pmatrix} 1.244 \\ 1.495 \end{pmatrix} \begin{pmatrix} 1.25 \\ 1.558 \end{pmatrix} \text{"Выход по R. Есть минимум."} \right]$$

::

****

<u>Вывод</u>: Точка минимума равна $XR1_{min} = \begin{pmatrix} 1.25 \\ 1.558 \end{pmatrix}$, значение функции в точке минимума равно $f(XR1_{min}) = 0.064$.

****

Рис. 2.60. Листинг программы расчета минимума функции Розенброка методом случайного поиска с возвратом при неудачном шаге (Пример 1.3, часть 5)

Аналитический расчет точки минимума функции

$$f\_R(x,y) := 100\cdot\left(y - x^2\right)^2 + (1 - x)^2$$

исследуемая функция (формальные параметры записаны в виде двух переменных)

1. Получить аналитические выражения производных фукции по переменным x ,y, используя символьный метод

$$\frac{d}{dx} f\_R(x,y) \rightarrow 2\cdot x - 400\cdot x\cdot\left(y - x^2\right) - 2 \qquad \frac{d}{dy} f\_R(x,y) \rightarrow 200\cdot y - 200\cdot x^2$$

2. Найти минимумы функций, используя блок расчета Given-MinErr. В качестве уравнений использовать выражения для первых производных (скопировав их).
При решении использовать метод LevenbergMarquardt.

$xx := 2 \qquad yy := 2$ начальные приближение

Given

$$2\cdot xx - 400\cdot xx\cdot\left(yy - xx^2\right) - 2 = 0 \qquad 200\cdot yy - 200\cdot xx^2 = 0$$

$xx \ge 0 \qquad yy \ge 0 \qquad yy \le 9 - xx$ Ограничения на переменные

$XT_{min} := \overrightarrow{Minerr(xx,yy)} \qquad XT_{min}^{\ T} = (1 \quad 1)$ координата точки минимума

$fXT_{min} := f\_R\left(XT_{min_0}, XT_{min_1}\right) = 0$ значение функции в точке минимума

Графическое отображение результатов расчета

Построить трехмерный график в виде контурных линий в диапазоне значений -6<x<6, -6<y<6, с шагом 0.1, нанести расчетные точки.

Расчет матрицы для постоения поверхности функции, используя дискретный аргумент.

Рис. 2.61. Листинг программы расчета минимума функции Розенброка методом случайного поиска с возвратом при неудачном шаге
(Пример 1.3, часть 6)

$x_{min} := -0.5 \quad x_{max} := 3 \qquad \Delta x := 0.02$

$y_{min} := -0.5 \quad y_{max} := 4 \qquad \Delta y := 0.02$

$Ni := \dfrac{x_{max} - x_{min}}{\Delta x} \qquad Ni = 175$

$Nj := \dfrac{y_{max} - y_{min}}{\Delta y} \qquad Nj = 225$ кол. точек расчета

$i := 0..Ni \qquad j := 0..Nj$

$x1_{i,j} := x_{min} + \Delta x \cdot i \quad y1_{i,j} := y_{min} + \Delta y \cdot j \qquad FM := \overrightarrow{f\_R(x1, y1)}$

Теоретические координаты точки минимума (красная точка)

$X12_0 := XT_{min_0} \qquad X12 = (1)$ значение х-составляющей

$Y12_0 := XT_{min_1} \qquad Y12 = ▮$ значение у-составляющей

$Z12_0 := fXT_{min} \qquad Z12 = ▮$ значение z-составляющей

Расчетные координаты точки минимума (черный крест)

$XT12_0 := XR1_{min_0} = 1.25$ значение х-составляющей

$YT12_0 := XR1_{min_1} = 1.558$ значение у-составляющей

$ZT12_0 := fXR1_{min} = 0.064$ значение z-составляющей

Координаты всех расчетных точек (черная кривая)

Преобразование расчетных точек в двухстолбцовый массив и расчет значений функции в этих точках. (Преобразование необходимы для построения точек на графике)

Рис. 2.62. Листинг программы расчета минимума функции Розенброка методом случайного поиска с возвратом при неудачном шаге
(Пример 1.3, часть 7)

X2z := submatrix(X1z ,0 ,last(X1z) − 1 ,0 ,0)   удаление из вектора последнего элемента

::

элементы вектора X2z с 0 по 6

X2z0_6 := submatrix(X2z ,0 ,5 ,0 ,0)

$$X1z0\_6^{T} = \left[\begin{pmatrix} 2 \\ 2 \end{pmatrix} \begin{pmatrix} 1.05 \\ 1.689 \end{pmatrix} \begin{pmatrix} 1.497 \\ 1.912 \end{pmatrix} \begin{pmatrix} 1.275 \\ 1.465 \end{pmatrix} \begin{pmatrix} 1.207 \\ 1.569 \end{pmatrix} \begin{pmatrix} 1.315 \\ 1.631 \end{pmatrix} \right]$$

элементы вектора X2z с 7 до конца

X2z7_end := submatrix(X2z ,6 ,last(X2z) ,0 ,0)

$$X2z7\_end^{T} = \left[\begin{pmatrix} 1.218 \\ 1.552 \end{pmatrix} \begin{pmatrix} 1.244 \\ 1.495 \end{pmatrix} \begin{pmatrix} 1.25 \\ 1.558 \end{pmatrix} \right]$$

::

X2G := v_k(X2z)

$$X2G^{T} = \begin{pmatrix} 2 & 1.05 & 1.497 & 1.275 & 1.207 & 1.315 & 1.218 & 1.244 & 1.25 \\ 2 & 1.689 & 1.912 & 1.465 & 1.569 & 1.631 & 1.552 & 1.495 & 1.558 \end{pmatrix}$$

$$fX2G := \overrightarrow{f\_R\left(X2G^{\langle 0 \rangle}, X2G^{\langle 1 \rangle}\right)}$$ значения функции в расчетных точках.

$$fX2G^{T} = (401 \quad 34.465 \quad 11.015 \quad 2.642 \quad 1.333 \quad 1.088 \quad 0.514 \quad 0.335 \quad 0.064)$$

Рис. 2.63. Листинг программы расчета минимума функции Розенброка методом случайного поиска с возвратом при неудачном шаге
(Пример 1.3, часть 8)

График поверхности исследуемой функции и полученных расчетных точек

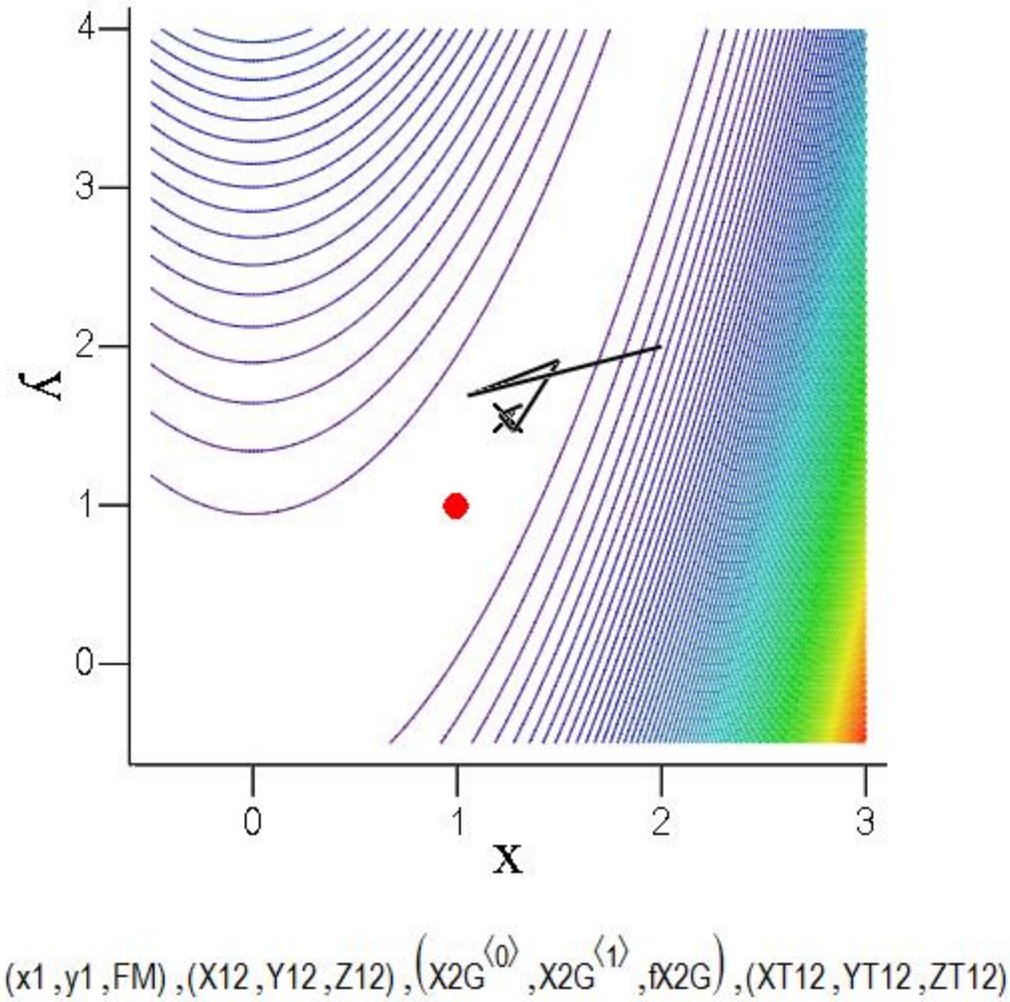
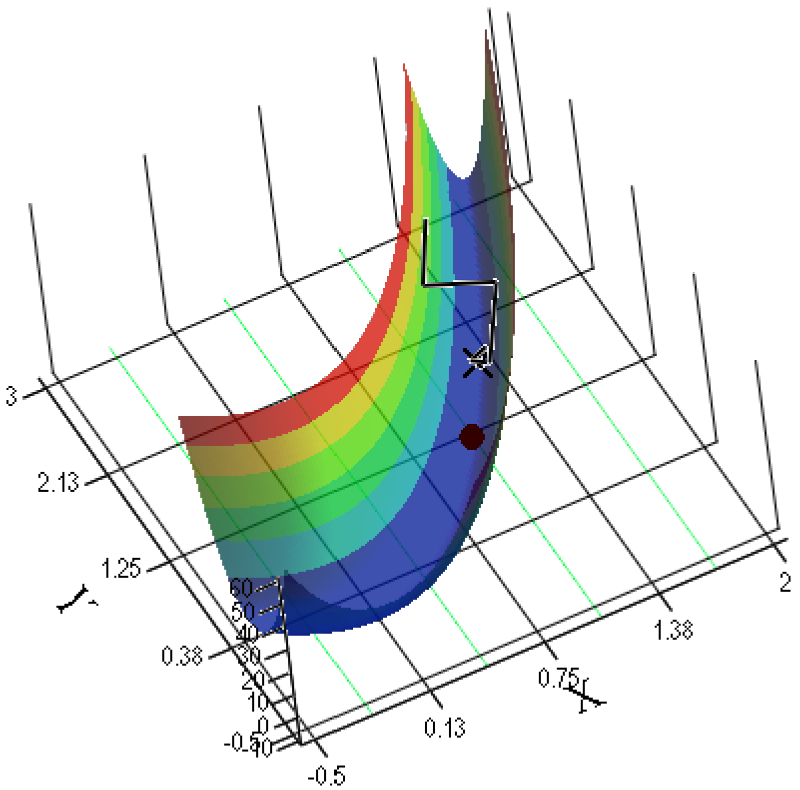


Рис. 2.64. Листинг программы расчета минимума функции Розенброка методом случайного поиска с возвратом при неудачном шаге
(Пример 1.3, часть 9)

Построить трехмерный график в виде поверхности и нанести стационарные точки (минимумы и максимумы).

Для получения вида поверхности приведенной на графике, необходимо установить диапазон значений по осям: x в итервале (-0.5 - 2, y - (-0.5 - 3), z - (-10 - 60).



$(x1, y1, FM), (X12, Y12, Z12), \left(X2G^{\langle 0 \rangle}, X2G^{\langle 1 \rangle}, fX2G\right), (XT12, YT12, ZT12)$

Рис. 2.65. Листинг программы расчета минимума функции Розенброка методом случайного поиска с возвратом при неудачном шаге
(Пример 1.3, часть 10)

СПИСОК ЛИТЕРАТУРЫ

1. Васильков Ю.В., Василькова Н.Н. Компьютерные технологии вычислений в математическом моделировании: Учеб. пособие. – М.: Финансы и статистика, 1999. – 256 с.

2. Дьяконов В.П. MathCAD 11/12/13 в математике: Справ. – М.: Горячая линия – Телеком, 2007. – 958 с.

3. Использование MathCAD в теории матриц: Метод. указания / И.В. Кудрявцева, В.А. Рыков, С.А. Рыков, С.В. Рыков. – СПб.: СПбГУНиПТ, 2011. – 50 с.

4. Охорзин В.А. Прикладная математика в системе MathCAD: Учеб. пособие. 2-е изд., испр. и доп. – СПб.: Лань, 2008. – 352 с.

5. Пантелеев А.В., Летова Т.А. Методы оптимизации в примерах и задачах: Учеб. пособие. 2-е изд., испр. – М.: Высш. шк., 2005. – 544 с.

6. Практические занятия в пакете MathCAD по исследованию систем линейных алгебраических уравнений: Пособие / В.А. Рыков, С.А. Рыков, И.В. Кудрявцева, С.В. Рыков. – СПб.: СПбГУНиПТ, 2009. – 107 с.

7. Реклейтис Г., Рейвиндран А., Рэгстел К. Оптимизация в технике. В 2 кн. Кн. 1. – М.: Мир, 1986. – 349 с.

8. Форсайт Дж., Малькольм Н., Моулер К. Машинные методы математических вычислений. –М.: Мир. –1980. –280 с.

9. Хаммельблау Д. Прикладное нелинейное программирование. – М.: изд. «МИР», 1975. – 534 с.

10. Рыков С.В., Рябова Т.В. Расчет линии фазового равновесия аммиака в пакете MathCAD // Холодильная техника и кондиционирование. 2013. № 2. С. 8.

11. Методы оптимизации в примерах в пакете MathCAD 15. Ч. I: Учеб. пособие / И.В. Кудрявцева, С.А. Рыков, С.В. Рыков, Е.Д. Скобов. – СПб.: НИУ ИТМО, ИХиБТ, 2014. – 166 с.

12. Методы оптимизации в примерах в пакете MathCAD 15. Ч. II: Учеб. пособие / И.В. Кудрявцева, С.А. Рыков, С.В. Рыков. – СПб.: НИУ ИТМО, ИХиБТ, 2015. – 178 с.

13. Практикум по работе в математическом пакете MathCAD: Пособие / С.В. Рыков, И.В. Кудрявцева, С.А. Рыков, В.А. Рыков – СПб.: НИУ ИТМО, ИХиБТ, 2015. – 84 с.

14. Рыков С.В., Камоцкий В.И., Рыков В.А. Расчет паровой ветви линии насыщения перфторпропана в пакете MathCad // Научный журнал НИУ ИТМО. Серия: Процессы и аппараты пищевых производств. 2014 № 1 С. 49.

15. Рыков С.В., Кудрявцева И.В., Киселев С.В. Расчет жидкостной ветви линии насыщения R218 в пакете MathCad // Научный журнал НИУ ИТМО. Серия: Холодильная техника и кондиционирование. 2014. № 1. С. 11.

16. Рыков С.В., Кудрявцева И.В., Рыков С.А., Рыков В.А. Методы оптимизации в примерах в пакете MathCAD 15. Часть 3. Многомерная оптимизация. Аналитические методы: Учебное пособие – Санкт-Петербург: Университет ИТМО, 2018. – 164 с.

17. Рыков С.В., Кудрявцева И.В., Рыков С.А., Рыков В.А. Методы оптимизации в примерах в пакете MathCAD 15. Часть 4. Методы оптимизации. Тесты с ответами: Учебное пособие – Санкт-Петербург: Университет ИТМО, 2018. – 85 с.

Рыков Сергей Владимирович
Рыков Сергей Алексеевич
Кудрявцева Ирина Владимировна
Старков Александр Сергеевич
Свердлов Александр Викторович

Методы оптимизации в примерах в пакете MATHCAD 15. Часть VI (Численный метод безусловной оптимизации нулевого прядка. Метод случайного поиска с возвратом при неудачном шаге)

Учебно-методическое пособие

В авторской редакции
Редакционно-издательский отдел Университета ИТМО
Зав. РИО Н.Ф. Гусарова
Подписано к печати
Заказ №
Тираж
Отпечатано на ризографе

Редакционно-издательский отдел
Университета ИТМО
197101, Санкт-Петербург, Кронверкский пр., 49