ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

«САМАРСКИЙ ГОСУДАРСТВЕНЫЙ УНИВЕРСИТЕТ»

Методы одномерной оптимизации

САМАРА 2015

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО

ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

«САМАРСКИЙ ГОСУДАРТСВЕННЫЙ УНИВЕРСИТЕТ»

Методы одномерной оптимизации

Методические указания

САМАРА

2015

УДК 510.2

Составители ***И.А. Власова***

Рецензент д-р.ф.-м.н.,.проф. А.Г.Коваленко..

Методы одномерной оптимизации: метод. указания / сост. *И.А. Власова* – Самара, 2015. –90 с.: ил.

Методические указания составлены для студентов по направлению подготовки «Прикладная математика и информатика», а также для других специальностей, в учебном плане которых имеется дисциплина «Методы оптимизации». В работе достаточно подробно рассмотрен раздел «Одномерная оптимизация» и даны методические рекомендации по использованию предложенной теории для решения задач одномерной оптимизация.

Методические указания также могут быть полезны при написании курсовых и дипломных работ для студентов других специальностей.\ Подготовлено на кафедре информатики и вычислительной математики.

УДК 510.2

Оглавление

Вводимые обозначения:

X – допустимое множество;

$x \in R^n$ - вектор, компоненты которого x_j, ($1 \le j \le n$)

$f(x)$, $x \in X$, - оптимизируемая или целевая функция;

x^*- точка оптимума(максимума или минимума);

$f(x^*)$-значение целевой функции в оптимальной точке;

$u(z)$- функция полезности;

$c=(c_1,c_2,\ \ , c_n)$- случайный вектор;

$\bar{c} = (\bar{c}_1, \ldots, \bar{c}_n)$***- математическое ожидание случайной величины;***

V- ковариационная матрица;

$A=[a_1,\ldots,a_n]$- матрица размера m x n, где a_j, ($1 \le j \le n$)-столбец матрицы A;

$z = (c,x)$ - общий доход.

Ведение

Динамику поведения любого физического объекта можно описать уравнением или системой уравнений, поэтому одной из основных задач исследователя является подбор функции, которая бы могла создать математическую модель реального объекта с заданной точностью при заданных граничных условиях. Подбор моделирующей функции тесно связан с некоторыми критериями или дополнительными условиями, выполнение которых сопровождается решением задачи оптимизации.

В теории оптимизации наряду с задачами оптимизации рассматриваются и другие задачи, такие как: задачи математического программирования, задачи вариационного исчисления, задачи исследования операций и др.

Задачей оптимизации называется задача о нахождении экстремума (минимума или максимума) вещественной функции в некоторой области. При этом рассматриваются области, заданные набором равенств и неравенств.

Математическое программирование изучает теорию и методы решения задач о нахождении экстремумов функций на множествах конечномерного векторного пространства, определяемых линейными и нелинейными ограничениями (равенствами и неравенствами).

Задачи линейного программирования были первыми, подробно изученными задачами поиска экстремума функций при наличии ограничений типа неравенств. В 1820 г. Ж. Фурье и затем в 1947 г. Дж. Данциг предложили метод направленного перебора смежных вершин в направлении возрастания целевой функции — симплекс-метод, ставший основным при решении задач линейного программирования.

Присутствие термина ***«программирование»*** объясняется тем, что первые исследования и первые приложения линейных задач оптимизации были в сфере экономики, и английское слово ***«programming»*** означает ***планирование***. Существующая терминология отражает тесную связь между математической постановкой задачи и её экономической интерпретацией.

Выделение класса экстремальных задач, определяемых линейным функционалом на множестве, заданном линейными ограничениями, состоялось в 30-х годах XX столетия. Первыми исследователями этих задач были: Джон фон Нейман, знаменитый математик и физик, доказавший основную теорему о матричных играх и изучивший экономическую модель, носящую его имя и советский академик, лауреат Нобелевской премии (1975 г.) Л. В. Канторович, сформулировавший ряд задач линейного программирования.

В 1931 г. венгерский математик Б. Эгервари рассмотрел математическую постановку задачи ***линейного программирования*** и решил её методом с названием ***«проблема выбора»***, который позднее получил название ***«венгерской метод».***

В 1939 году Л. В. Канторович предложил метод разрешающих множителей, незначительно отличающийся от симплекс-метода, а в 1949 году совместно с М. К. Гавуриным разработал ***метод потенциалов***, который применяется для решения транспортных задач.

Дальнейшие исследования математической теории линейного и нелинейного программирования представлены в работах Л. В. Канторовича, В. С. Немчинова, В. В. Новожилова, А. Л. Лурье, А. Брудно, А. Г. Аганбегяна, Д. Б. Юдина, Е. Г. Гольдштейна и других математиков и экономистов.

Одновременно с развитием линейного программирования большое внимание уделялось задачам нелинейного программирования, в которых либо целевая функция, или ограничения, либо то и другое были нелинейными. В 1951 г была опубликована работа Куна и Таккера, в которой приведены необходимые и достаточные условия оптимальности для решения задач нелинейного программирования. Эта работа послужила основой для последующих исследований в этой области.

Интенсивное развитие методов оптимизации и математического программирования началось с появлением электронных вычислительных машин. Начиная с 1955 года, опубликовано большое количество работ, посвященных квадратичному программированию, это работы: Била, Э. Баранкина (Barankin E.) и Дорфмана (Dorfman R.), Франка (Frank M.) и Вольфа (Wolfe P.), Г. Марковица и др.

В это же время продолжилось развитие градиентных методов, предназначенных для решения задач нелинейного программирования. Этим методам посвящены работы Денниса (Dennis J. B.), Розена (Rosen J. B.) и Зонтендейка (Zontendijk G.)

В процессе моделирования объектов ставится задача определения наилучшей структуры и соответствующих значений параметров объектов. Такая задача называется *оптимизационной*. Если оптимизация связана с расчетом оптимальных значений параметров при заданной структуре объекта, то она называется *параметрической*. Задача выбора оптимальной структуры называется *структурной оптимизацией*.

Учебное пособие содержит основные положения теории оптимизации и численные методы решения соответствующих задач. Объектом изучения являются стационарные детерминированные системы. К ним относятся: целевые организационные системы, экономические, проектные, конструкторские и другие системы, не изменяющиеся во времени и не зависящие от случайных факторов. Предметом изучения является проблема оптимизации систем, имеющих три аспекта: конструктивный, структурный и функциональный.

Конструктивная оптимизация заключается в повышении производительности и ресурсов системы за счет выбора оптимальных параметров её элементов и каналов связи. Такие задачи, как правило, являются непрерывными и опираются на физические законы.

Структурная оптимизация заключается в минимизации стоимости системы за счет сокращения числа её элементов и совершенствования взаимодействия между ними. Задачи структурной оптимизации в большинстве случаев являются дискретными и носят комбинаторный характер. Они опираются на теорию систем, комбинаторику, теорию графов и другие науки.

Функциональная оптимизация – это оптимальное планирование и управление, т.е. повышение эффективности системы путем оптимизации режимов её функционирования. Такие задачи могут быть как непрерывными, так и дискретными. Они опираются на общественные и естественные науки.

Методом изучения задач оптимизации является математическое моделирование. В качестве моделей используются линейные и нелинейные задачи оптимизации.

Исследование свойств задач оптимизации и разработка методов их решения производятся на основе теории выпуклых функций, выпуклых множеств, линейной алгебры и теории матриц.

В методическом пособии рассмотрены задачи и методы условной и безусловной оптимизации, которые представляют не только большой самостоятельный интерес, но и служат основой для решения проблемы оптимизации в целом. Так, например, одномерные задачи безусловной оптимизации помогают испытывать новые идеи и методы, исследовать вопросы сходимости и устойчивости алгоритмов, искать подходы к решению многоэкстремальных задач.

Методические указания разработаны на основе лекций, прочитанных на механико-математическом факультете и на факультете экономики и управления Самарского государственного университета. Оно содержит базовые сведения теории оптимизации. Многие модели и методы проиллюстрированы примерами задач с решениями. Для закрепления изученного материала предложены варианты лабораторных работ для самостоятельного выполнения.

Пособие предназначено для студентов, обучающихся по специальностям «Прикладная математика и информатика», «Математические методы в экономике», а также может быть полезна студентам других специальностей при изучении курса «Методы оптимизации».

1. Задачи оптимизации. Классификация. Локальный и глобальный экстремум.

Стандартная математическая задача оптимизации формулируется следующим образом:

среди элементов x, образующих множество X, найти такой элемент x*, который доставляет минимальное (или максимальное) значение f(x*) заданной функции f(x).

Для того, чтобы *корректно поставить задачу оптимизации* необходимо задать:

- допустимое множество: $(X=\{x=(x_1, x_2, \ldots, x_n) \mid g_i(x) \leq 0, i=1,2, \ldots, m\} \subset R^n)$;
- целевую функцию — отображение $(f(x),), X \rightarrow R$;
- критерий поиска - *max* или *min*.

Решить задачу оптимизации $f(x) \rightarrow min\ (max)$ означает одно из следующих действий:

- показать, что $X = \varnothing$;
- показать, что целевая функция *f(x) не ограничена снизу (сверху);*
- найти такое $x^* \in X$, чтобы $f(x^*) = min\ (max)\ f(x)$

Если функция, для которой ищется оптимальное значение, не является выпуклой, то ограничиваются поиском локальных минимумов или максимумов: точек x^* таких, что всюду в некоторой их окрестности $f(x) \geq f(x^*)$ для минимума и $f(x) \leq f(x^*)$ для максимума.

Если *допустимое множество* $X=R^n$, то такая задача называется задачей *безусловной оптимизации,* если *допустимое множество* $(X=\{x=(x_1,x_2,\ldots,x_n) \mid g_i(x) \leq 0, i=1,2,\ldots,m\} \subset R^n)$, то задача называется *задачей условной оптимизации.*

Общая запись задач оптимизации порождает разнообразие их классов. От класса задачи зависит подбор методов, от которых зависит эффективность её решения. Классификацию задач определяют целевая функция и допустимая область, которая задается системой равенств и неравенств или более сложными алгоритмами.

Классификация. Существуют различные способы классификации задач оптимизации и соответствующих методов их решения. Рассмотрим некоторые из них.

По видам экстремумов задачи оптимизации классифицируются на:

– задачи поиска локальных экстремумов, им соответствуют *локальные методы*;

– задачи поиска глобальных экстремумов, им соответствуют *глобальные методы*.

Локальные методы сходятся к какому-нибудь локальному экстремуму целевой функции. В случае унимодальной целевой функции, этот экстремум единственен, и будет глобальным максимумом или глобальным минимумом.

Глобальные методы имеют дело с многоэкстремальными целевыми функциями. При глобальном поиске основной задачей является выявление тенденций глобального поведения целевой функции.

Существующие методы поиска можно разбить на три большие группы: *детерминированные, случайные (или стохастические) и комбинированные.*

По критерию размерности допустимого множества, методы оптимизации делят на *методы одномерной оптимизации* и *методы многомерной оптимизации*.

По виду целевой функции и допустимого множества, задачи оптимизации и методы их решения можно разделить на следующие классы:

1. *Задачи линейного программирования* – это задачи оптимизации, в которых целевая функция $f(x)$ и ограничения, где $x=(x_1, x_2, \ldots, x_n)$, $i=1,2, \ldots, m$, являются линейными. Для решения таких задач используются методы линейного программирования.

2. *Задачи нелинейного программирования* - это задачи оптимизации, в которых либо целевая функция $f(x)$, $x=(x_1, x_2, \ldots, x_n)$, либо ограничения (или хотя бы одно из ограничений), $g_i(x)$, $i=1,2, \ldots, m)$ являются нелинейными функциями.

Среди задач *нелинейного программирования*, в свою очередь, выделяют две частные задачи:

а) если целевая функция $f(x)$, $x=(x_1, x_2, ..., x_n)$ и ограничения задачи $g_i(x)$, $(i=1,2,...,m)$—выпуклые функции, то такую задачу называют задачей *выпуклого программирования*;

б) если значения искомых параметров дискретные ($X \subset Z$), то имеют дело с задачей *дискретного программирования.*

По требованиям к гладкости и наличию у целевой функции частных производных методы оптимизации можно разделить на:

– *прямые методы (или методы нулевого порядка),* требующие только вычислений целевой функции в точках приближений и не требующие вычисления производных;

– *методы первого порядка,* требующие вычислений целевой функции в точках приближений *и первых частных производных функции;*

– *методы второго порядка,* требующие вычислений целевой функции в точках приближений а также *первых и вторых частных производных, то есть гессиана целевой функции.*

В отдельных случаях, в научных исследованиях, могут быть использованы методы более высоких порядков, однако на практике методы выше второго порядка не используются в связи со значительным усложнением вычислительного процесса.

Кроме названной классификации, оптимизационные методы делятся на *аналитические, численные и графические.*

Исследование множества *X* позволило классифицировать задачи математического программирования в *зависимости от природы этого множества* следующим образом:

– *задачи дискретного программирования (или комбинаторной оптимизации)* — если X конечно или счётно;

– *задачи целочисленного программирования* — если X является подмножеством множества целых чисел;

– *задачи нелинейного программирования,* если хотя бы одно ограничение или целевая функция содержат нелинейные функции и X является подмножеством конечномерного векторного пространство;

– *задачи линейного программирования,* если все ограничения и целевая функция содержат лишь линейные функции.

Способ нахождения экстремума полностью определяется классом задачи. Но перед тем как получить математическую модель, нужно выполнить 4 этапа моделирования.

Этап 1. Определение границ системы оптимизации. Отбрасываем те связи объекта оптимизации с внешним миром, которые не могут сильно повлиять на результат оптимизации, а, точнее, те, без которых решение упрощается.

Этап 2. Выбор управляемых переменных. «Замораживаем» значения некоторых переменных (неуправляемые переменные). Другие оставляем принимать любые значения из области допустимых решений (управляемые переменные).

Этап 3. Определение ограничений на управляемые переменные (равенства и\или неравенства).

Этап 4. Выбор числового критерия оптимизации. Создание целевой функции.

1.1. Экстремальные задачи и их свойства

Многие задачи, как практического, так и теоретического характера касаются выбора «наилучшей» конфигурации или множества параметров для достижения некоторой цели. Сложилась иерархия таких задач вместе с соответствующим набором методов их решения. Главным объектом иерархии является общая задача нелинейного программирования (НЛП):

$$f(x) \to \min, \tag{1.1.1}$$

$$g_i(x) \le 0, \quad i=1,2,\ldots\ldots r \tag{1.1.2}$$

$$g_i(x) = 0, \quad i=r+1, r+2, \ldots.m \tag{1.1.3}$$

где $f(x)$, $g(x)_i$ – произвольные функции параметра x, $x \in R^n$.

Производя замену $\bar{f}(x) = -f(x)$, перейдем от задачи минимизации к задаче максимизации. Поэтому рассмотренные в пособии методы можно использовать как для решения задач минимизации, так и для решения задачи максимизации.

В задаче (1.1.1) - (1.1.3) $f(x)$ - целевая функция, а x - множество точек ($x \in R^n$, $R^n \to R^1$), которые удовлетворяют ограничениям (1.1.2), (1.1.3) . Ограничения (1.1.2), (1.1.3) являются допустимым множеством решения задачи оптимизации, будем обозначать его X.

В теории математического программирования изучаются: проблемы существования решения, проблемы формирования системы оптимальных признаков, методы вычисления оптимальных решений и т.п. В качестве примеров рассмотрим некоторые задачи математического моделирования, в которых методы оптимизации используются для определения оптимальных параметров модели, и, следовательно, её оптимальной структуры.

Пример 1. Задача оценки параметров и структуры математической модели.

На практике в процессе разработки математических моделей возникают задачи поиска оптимальной структуры модели. Например, если для изучения какого-либо явления конструируется математическая модель, то к оптимизации прибегают для того, чтобы определить ее структуру и параметры, которые обеспечивают наилучшее согласование модели с реальностью.

Пусть результат измерения случайной величины y зависит от $x \in R^m$, причем

$$E(y/x) = \eta(x,\theta), \tag{1.1.4}$$

где y - результат измерения, $\eta(x,\theta)$ – функция, характеризующая математическую модель, вид которой известен, $\theta \in R^m$ - неизвестные параметры функции.

Оценка неизвестных параметров θ при определенных условиях может быть произведена, например, по методу наименьших квадратов посредством минимизации суммы квадратов отклонений по θ.:

$$S(\theta)=\sum_{i=1}^{N}\left[y_i-\eta(x_i,\theta)\right]^2 \to \min \tag{1.1.5}$$

здесь N- количество наблюдений.

Пример2. Задача распределения ресурсов в условиях не полной информации

Пусть имеется m ресурсов в количествах, задаваемых компонентами b_j , $(1 \le j \le m)$, вектора $b \in R^m$. Обозначим через $A=[a_1,\dots,a_n]$ матрицу размера m x n. Столбец a_j, $(j=1,2, \dots,n)$ – матрицы A характеризует затраты ресурсов на единицу интенсивности j-го способа производства.

Обозначим через $x \in R^n$ вектор, компоненты которого x_j , $(1 \le j \le n)$, продукция j-го способа производства, а через $c \in R^n$ вектор, компоненты которого c_j, $(1 \le j \le n)$, доход от единицы продукции j-го способа производства. Тогда $z = (c,x)$ составляет общий доход.

Таким образом, в принятых обозначениях задача распределения ресурсов между способами производства с целью максимизации дохода имеет вид:

$$\begin{aligned} &z = (c\ x) \to \max, \\ &Ax \le b, \\ &x \ge 0. \end{aligned} \tag{1.1.6}$$

Для некоторых практических задач такая детерминированная модель не адекватна реальности, так как c_j, $(1 \le j \le n)$ случайные величины. В этом случае для построения математической модели используется не сам вектор c_j, $(1 \le j \le n)$,а его математическое ожидание. Предположим, что c - случайный вектор с математическим ожиданием $\overline{c} = (\overline{c}_1,\dots,\overline{c}_n)$ и ковариационной матрицей V. Тогда значение целевой функции z будет случайной величиной с математическим ожиданием $(\overline{c},x)$ и дисперсией (x,Vx). Тогда задача максимизации дохода принимает вид:

$$\begin{aligned} &(\overline{c},x) \to \max, \\ &Ax \le b, \\ &x \ge 0. \end{aligned} \tag{1.1.7}$$

Если требуется минимизировать дисперсию показателя z, то необходимо решить задачу в виде:

$$\begin{aligned} &(x,Vx) \to \min, \\ &Ax \le b \\ &x \ge 0, \end{aligned} \tag{1.1.8}$$

здесь V- ковариационная матрица.

В реальных задачах возникает потребность получения максимального дохода при малых значениях дисперсии. Такие задачи называются многокритериальными. Для некоторых задач постановка многокритериальной задачи может быть сведена к однокритериальной. Например, требуется достигнуть минимума:

$$\begin{aligned} &(x, Vx) \to \min, \\ &Ax \le b, \\ &(\bar{c},x) \ge \bar{Z} \\ &x \ge 0. \end{aligned} \tag{1.1.9}$$

Другой подход может состоять в минимизации вероятности недостижения заданного уровня дохода $Q=P\{(c,x) \ge Z\}$. Пусть $c = d+yf$, где d, f - детерминированные векторы, а y- случайная составляющая, тогда

$$\alpha = P\{(d,x) + y(f,x) \ge \bar{z}\} = P\left\{ y \ge \frac{\bar{Z} - (d,x)}{(f,x)} \right\} \tag{1.1.10}$$

Тогда, максимизация α сводится к решению задачи:

$$\begin{aligned} &\frac{\bar{Z} - (d,x)}{(f,x)} \to \min, \\ &Ax \le b, \\ &x \ge 0. \end{aligned} \tag{1.1.11}$$

Если c - случайный вектор, то в зависимости от степени неприятия риска k производится максимизация ожидаемого значения полезности:

$$\begin{aligned} &k(\bar{c},x) - \frac{1}{2} q^2 x^T V x \to \max, \\ &Ax \le b, \\ &x \ge 0. \end{aligned} \tag{1.1. 12}$$

Приведенная функция полезности учитывает степень риска, связанного со случайным характером величины вектора дохода c, которая основана на функции полезности индивида при условии нормального закона распределения вектора c и имеет вид:

$$\boldsymbol{u(z) = 1 - e^{-qz}} \tag{1.1.13}$$

Здесь $\boldsymbol{q}$ – параметр нормального распределения.

Функция полезности индивида при увеличении или уменьшении дохода z на величину Δz более существенно реагирует на уменьшение дохода.

1.2. Локальный и глобальный экстремум. Существование решения

В задаче (1.1.1) – (1.1.3) различают точки минимума двух видов: точки локального минимума и точка глобального минимума.

Определение 1.2.1. *Точка $x^* \in X$ называется точкой* ***локального минимума****, если $f(x^*) \le f(x)$ $\forall x \in O\{\varepsilon(x^*)\}$, где $O\{\varepsilon(x^*)\} = \{x \in Rn \mid //x^* - x// \le \varepsilon\}, \varepsilon$ - окрестность точки $x^*, \varepsilon > 0$.*

Определение 1.2.2. *Точка x^* называется точкой* ***глобального максимума****, если $f(x^*) \ge f(x)$ $\forall x \in X$.*

Определение 1.2.3 *.Множество $x \in Rn$ называется* ***компактным****, если любая последовательность $\{x_k\} \in X$ имеет, хотя бы одну предельную точку $x^* \in X$.*

Известно, что всякая ограниченная последовательность имеет хотя бы одну предельную точку. Поэтому в R_n ***компактным*** является любое замкнутое ограниченное множество.

Достаточные условия существование оптимального решения задачи (1.1.1) - (1.1.2) сформулированы в теореме 1.2.1.

Теорема 1.2.1. *(Вейерштрасса). Для того чтобы в задаче (1.1.1)-(1.1.3) существовала точка глобального минимума, достаточно, чтобы допустимое множество X было компактно, а целевая функция f(x) непрерывна на X.*

В силу сложности проверки ограниченности множества X, а зачастую, в силу его неограниченности на практике часто применяется следствие теоремы Вейерштрасса.

Следствие 1.2.1. *Если функция* $f(x)$ *непрерывна в* R_n *и* $\lim\limits_{||x||\to\infty} f(x) = +\infty$, *то* $f(x)$ *достигает своего глобального минимума в любом замкнутом подмножестве* R_n.

2. Одномерная оптимизация

2.1. Задача одномерной оптимизации. Классификация экстремумов

Задача одномерной оптимизации заключается в определении экстремумов непрерывной функции одной переменной $\varphi(x)$ в замкнутом промежутке $E=[a,b]$, который может быть и неограниченным. Такие задачи возникают при исследовании экстремальных свойств характеристик одномерных процессов и систем, при анализе отдельных факторов многомерных процессов и систем, при редукции многомерных задач к одномерным и в других случаях.

Многие методы многомерной оптимизации на некотором *k-ом* шаге сводятся к выбору направления S_k в точке x_k и к поиску экстремума на луче (2.1.1), т. е. к одномерной задаче

$$x(\lambda) = x_k + \lambda S_k \qquad (2.1.1.)$$

Здесь λ определяет величину шага поиска по направлению S_k .

Методы одномерной оптимизации представляют собой не только большой самостоятельный интерес, но и служат основой успешного решения проблем оптимизации в целом. Одномерные задачи являются своего рода эталоном, на котором испытываются новые идеи и методы оптимизации, исследуются вопросы сходимости и устойчивости, ищутся подходы к решению задач и т. д.

Определение 2.1.1. *Любой экстремум* E_k^* *функции* $\varphi(x)$ *на множестве* $E=[a,b]$ *определяется оптимальным значением функции* z^*_k *и множеством оптимальности аргумента* x_k^*: $E_k^* = \{ z^*_k , x_k^*\}$ *(см. рис.2.1.1.).*

Он характеризуется следующими свойствами:

1. $X_k^*= [a_k^*, b_k^*]$-связное множество, Z_k^*- константа;

2. $\varphi(x)=Z_k^*$-*для всех* $x \in X_k^*$; (2.1.2)

3. $\varphi(x)> Z_k^*$, либо $\varphi(x)< Z_k^*$ для всех $x \in O(X_k^*/E)$.

Здесь $O(X_k^*/E)$ – некоторая окрестность множества X_k^* в пределах множества E, для которой:

$$O(X_k^*) \Lambda E, \ X_k^* \not\subset O(X_k^*/E) \qquad (2.1.3)$$

Экстремум является минимумом, если в третьем свойстве (2.1.2) определения (2.1.1) выполняется первое условие, и максимумом, если выполняется второе условие.

Постоянная функция $Z=const$ не имеет ни максимума, ни минимума.

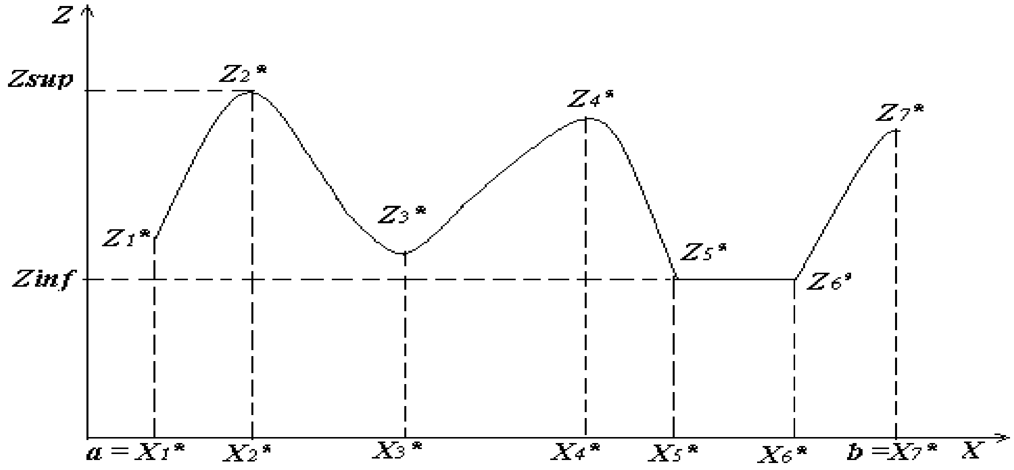

Рис. 2.1.1. Экстремумы функции ***φ(x).***

В случае неограниченного промежутка $[a,b]$ это множество может быть неограниченным слева или справа. При этом, множества оптимальности различных экстремумов $E_k^*=\{z^*_k , x_k^*\}$ и $E_s^*=\{ z^*_s , x_s^*\}$ не пересекаются : $x_k^* \Lambda x_s^* =\emptyset$. Однако значения x_k^* и x_s^* могут и совпадать.

Если $a_k^*= b_k^*= x_k^*$, то экстремум $\{ z^*_k , x_k^*\}$ является ***точечным.***

Если же $a_k^*< b_k^*$, то экстремум $\{ z^*_k , [a_k^*, b_k^*]\}$ назовем ***интервальным.***

Если множество оптимальности X_k^* содержит одну из концевых точек a или b, то экстремум будет ***граничным,*** а в противном случае – ***внутренним.***

Граничный экстремум окаймляется окрестностью $O(X_k^*/E)$ с одной стороны, а ***внутренний*** экстремум - с обеих сторон. ***Граничные точки*** a и b являются ***экстремальными.***

Определение 2.1.2. *Экстремум* называется *отделимым*, если существует такая окрестность $O(X_k/E)$, в которой нет точек других экстремумов.

Определение 2.1.4. *Экстремум* называется *неотделимым* или *предельным*, если не существует такой окрестности $O(X_k/E)$, в которой нет точек других экстремумов.

Большое разнообразие экстремумов вызывает необходимость классификации экстремумов. Все характеристики экстремумов сведем в таблицу 2.1.1.

Таблица 2.1.1.

№	***Характеристика***	***Вид экстремума***	
		Минимум	***Максимум***
1.	***Характер***	*Локальный*	*Локальный*
		Глобальный	*Глобальный*
2.	***Структура***	*Точечный*	*Точечный*
		Интервальный	*Интервальный*
3.	***Положение***	*Внутренний*	*Внутренний*
		Граничный	*Граничный*
4.	***Отделимость***	*Отделимый*	*Отделимый*
		Неотделимый	*Неотделимый*

Для экстремумов, представленных на рисунке 2.1.1. классификация экстремумов имеет вид (таблица 2.1.2).

Таблица 2.1.2.

№	***Экстремумы***		***Класс экстремумов***
	z^*_k	x_k^*	
1	z^*_1	x_1^*	*Граничный минимум*
2	z^*_2	x_2	*Глобальный максимум*
3	z^*_3	x_3^*	*Локальный минимум*
4	z^*_4	x_4^*	*Локальный максимум*

5	z^*_5	x_5^*	*Интервальный минимум, глобальный*
6	z^*_6	x_6^*	*Интервальный минимум, глобальный*
7	z^*_7	x_7^*	*Граничный максимум*

Совокупность всех экстремумов функции *φ(x) на отрезке [a,b] образуют спектр*. Проведенный анализ показывает, что проблема одномерной оптимизации является достаточно сложной задачей.

Изучение каждого класса экстремумов требует индивидуального подхода. Решая задачу, мы не знаем заранее, к какому классу относится исследуемая функция, поэтому нужно быть готовым к любому варианту. Обычно налагают довольно жесткие ограничения, например, требуют единственность экстремума. Это приводит к типовым задачам одномерной оптимизации, которые представлены в таблице 2.1.1.

Разрешимость всех типовых задач для непрерывной функции *φ(x)* на ограниченном замкнутом промежутке *[a,b]* гарантируется теоремой Вейерштрасса. Если же промежуток *[a,b]* не ограничен, то функция *φ(x)* может достигать экстремума при $x^*=\pm\infty$, причем этот экстремум может быть также неограниченным: $z^* = \pm\infty$. Такие экстремумы называются несобственными.

2.2. Методы решения задач одномерной оптимизации.

Для решения задач одномерной оптимизации разработаны специальные методы. Они подразделяются на *методы локальной и глобальной оптимизации*.

Методы локальной оптимизации предназначены для поиска единственного точечного локального экстремума. Они классифицируются, как и все методы оптимизации, по порядку используемых в них производных целевой функции *φ(x)*.

К *методам нулевого порядка* относятся методы случайного поиска, сканирования, оптимального поиска и другие. Наиболее эффективными из них являются методы оптимального поиска.

К *методам первого порядка* относятся: метод деления отрезка пополам (метод Больцано), метод касательных и другие.

К *методам второго порядка* принадлежат: методы Ньютона и методы квадратичной аппроксимации и интерполяции.

К методам глобальной оптимизации относятся методы, предназначенные для решения многоэкстремальных задач. Основным и наиболее часто используемым методом является *метод глобального поиска*. Алгоритм этого метода рассмотрен в параграфе 2.8.

Вопросы сходимости и устойчивости алгоритмов одномерной оптимизации рассмотрены в работах Беллмана [2], Уайлда [10], Карманова [4] и других авторов.

2.3. Сходимость и устойчивость алгоритмов одномерной оптимизации

Практическое применение многих методов оптимизации обнаруживает их неустойчивость, которая выражается в том, что в ходе проведения итераций погрешности вычислений возрастают. В результате алгоритмы не только перестают сходиться, но даже начинают осциллировать в неограниченных пределах.

В методах оптимизации вычислительные погрешности принято интерпретировать, как возмущения некоторого параметра, характеризующего алгоритмы. Рассмотрим основные определения и теоремы, на которых базируются алгоритмы одномерной оптимизации.

2.3.1. Основные определения

Пусть некоторый алгоритм A действует в n-мерном евклидовом пространстве Rn и по заданному начальному приближению $x_0 \in Rn$ вырабатывает бесконечную последовательность $x_k = x_k(x_0)$, k=1,2, .. .

Определение 2.3.1.. *Будем говорить, что алгоритм А сходится при начальном приближении $x_0 \in Rn$, если он порождает сходящуюся последовательность:*

$$\lim x_k(x_0) = x(x_0) \tag{2.3.1}$$

Если алгоритм A порождает систему вложенных промежутков $[a_k, b_k] \subset [a_{k-1}, b_{k-1}]$, $k=1,2,\ldots$, то $x_k \in (a_k, b_k)$ и сходимость алгоритма A означает, что последовательность промежутков $\{[a_k, b_k]\}$ стягивается к некоторому промежутку $[a, b]$, причем $a_k \nearrow a$, $b_k \searrow b$.

Пусть теперь алгоритм A зависит от параметра λ, , т.е. $A=A(\lambda)$, $\lambda \in Rm$, тогда и элементы последовательности $\{x_k\}$, порождаемой этим алгоритмом, будут зависеть от λ: $x_k=x_k(\lambda)=x_k(\lambda,x_0)$.

Определение 2.3.2. *Алгоритм $A(\lambda)$ является непрерывным по λ в точке $\lambda_0 \in Rm$, если все элементы последовательности $\{x_k(\lambda)\}$ непрерывны в точке λ_0.*

Определение 2.3.3 *Алгоритм $A(\lambda)$ назовем устойчивым при $\lambda=\lambda_0$, если элементы порождаемой им последовательности $\{x_k(\lambda)\}$ непрерывны по λ в точке λ_0 и равномерны относительно k, $k=1,2,\ldots$.*

Суть устойчивости заключается в том, что устойчивый алгоритм $A(\lambda)$ для близких значений λ и λ_0, параметра λ порождает близкие в целом последовательности $\{x_k(\lambda)\}$, $\{x_k(\lambda_0)\}$.

Из определения не следует, что устойчивость алгоритма эквивалентна его сходимости: сходящийся алгоритм может быть неустойчивым и, наоборот, расходящийся алгоритм может обладать устойчивостью.

Например, арифметическая прогрессия. Алгоритм непрерывно зависит от двух параметров a и d : $a_k=a_{k-1}+d = a + kd$. . Алгоритм расходится, но является устойчивым при любых значениях параметров a и d.

Определение 2.3.4. *Алгоритм $A(\lambda)$ назовем асимптотически устойчивым при $\lambda=\lambda_0$, если для заданного $\varepsilon>0$ найдется такое $\delta=\delta(\varepsilon)>0$ и такой номер $k_0=k_0(\varepsilon,\delta)$, что*

$$\| x_k(\lambda)-x_k(\lambda_0)\|<\varepsilon,\ \forall k>k_0, \forall\lambda: \|\lambda - \lambda_0\|<\delta \qquad (2.3.2)$$

<u>Замечание.</u> Если алгоритм $A(\lambda)$ устойчив, то он и асимптотически устойчив. Обратное утверждение неверно.

2.3.2. Основные теоремы

1) Теорема об устойчивости алгоритма

Теорема 2.3.1. *(об устойчивости алгоритма). Пусть алгоритм* $A(\lambda)$ *вырабатывает последовательность* $\{x_k(\lambda)\}$ *в евклидовом пространстве* Rn *по начальному приближению* x_0 *. Тогда из непрерывности и равномерной сходимости алгоритма* $A(\lambda)$ *в области* Ω *следует его устойчивость в этой области.*

<u>*Доказательство.*</u>

Пусть алгоритм $A(\lambda)$ является непрерывным в области Ω.

Тогда предел $x(\lambda)=\lim x_k(\lambda)$, $(\lambda\in\Omega\subset Rm, k\to\infty)$ равномерно сходящейся последовательности будет также непрерывным и, следовательно, при любых $\lambda,\lambda_0 \in \Omega$ справедливо неравенство

$$//x_k(\lambda)-x_k(\lambda_0)//\leq /x_k(\lambda)-x(\lambda)//+//x_k(\lambda_0)-x(\lambda_0)//+//x(\lambda)-x(\lambda_0)// \qquad (2.3.3)$$

В силу равномерной сходимости $\{x_k(\lambda)\}$ по заданному $\varepsilon>0$ найдутся такие $\delta_0=\delta(\varepsilon)>0$ и $k_0=k_0(\varepsilon,\delta)$, что $//x_k(\lambda)-x(\lambda)// <\varepsilon/3,\ \forall k>k_0, , \forall\lambda: //\lambda - \lambda_0//<\delta_0$

Тогда из непрерывности предела $x(\lambda)=\lim x_k(\lambda)$, $\lambda\in\Omega\subset Rm$, следует, что

$$//x(\lambda)-x(\lambda_0)//<\varepsilon/3, \text{ если } //\lambda - \lambda_0//<\delta_1$$

Если же $k<k_0$, то мы имеем лишь конечное число элементов $x_1(\lambda)$, $x_2(\lambda)$, ... ,$x_n(\lambda)$, непрерывных по λ, поэтому

$$//x_k(\lambda)-x_k(\lambda_0)//<\varepsilon/3, \forall k=1,2,\dots,k_0, \forall\lambda: //\lambda - \lambda_0//<\delta_2.$$

Сопоставляя все полученные оценки, видим, что $//x_k(\lambda)-x_k(\lambda_0)// <\varepsilon$ для $\forall k=1,2,\dots,k_0$, $\forall\lambda: //\lambda - \lambda_0//<\delta$, где $\delta= \min(\delta_0, \delta_1, \delta_2)$. Таким образом, устойчивость алгоритма доказана.

2). Теорема о непрерывности алгоритма

Теорема 2.3.2. *(о непрерывности алгоритма). Если алгоритм* $A(\lambda)$ *сходится и устойчив в некоторой точке* $\lambda_0\in R_n$ *, то его предел* $x(\lambda)=\lim x_k(\lambda)$ *является непрерывным в этой точке.*

<u>*Доказательство.*</u>

В силу устойчивости алгоритма следует, что для заданного $\varepsilon>0$ найдется такие $\delta>0$, что

$$\| x_k(\lambda)-x_k(\lambda_0)\| <\varepsilon,\ \forall k=1,2,\ldots,k_0,\ \text{,если}\ \|\lambda - \lambda_0\|<\delta. \qquad (2.3.4)$$

а это означает непрерывность $x(\lambda)$ в точке λ_0.

Из доказанной теоремы видно, что устойчивость алгоритма обеспечивает непрерывную зависимость предела $x(\lambda)=\lim x_k(\lambda)$ от параметра λ. Погрешности вычислений часто интерпретируются как возмущения некоторого параметра λ и гарантирует, что малым погрешностям $\Delta\lambda=\lambda-\lambda_0$ будут соответствовать малые отклонения предела $\Delta x(\lambda) = x(\lambda)-x(\lambda_0)$.

2.4. Числа Фибоначчи

В алгоритмах оптимального поиска для определения параметра λ используются числа Фибоначчи. Числа Фибоначчи представляют собой элементы числовой последовательности *0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765, 10946, …*, в которой каждое последующее число равно сумме двух предыдущих чисел. Эти числа сначала изменяются медленно, а затем стремительно возрастают, так, например, уже F_{30} будет больше 10^6 (см. таблицу 2.4.1) .

Свое название эти числа получили по имени средневекового математика Леонардо Пизанского (известного как Фибоначчи). Более формально, последовательность чисел Фибоначчи $\{F_n\}$ задается линейным рекуррентным соотношением:

$$F_0 = 0,\ F_1 = 1,\ F_n = F_{n+2} - F_{n+1.},\ n\geq 2 \qquad (.2.4.1)$$

Иногда число *0* не рассматривается как член последовательности.

Числа Фибоначчи могут быть рассмотрены и для отрицательных номеров *n* как двусторонне бесконечная последовательность, удовлетворяющая тому же рекуррентному соотношению. При этом члены с отрицательными индексами легко получить с помощью эквивалентной формулы «назад»:

$$F_n = F_{n+2} - F_{n+1.}$$

Числа Фибоначчи можно найти и в аналитической форме в зависимости от k. Определим фундаментальное решение системы (2.4.1), полагая $F_k = r^{k/}$. Тогда из (2.4.1) получим уравнение

$$r^2 = r + 1 \qquad (2.4.2)$$

Его корнями служат

$$r_1 = \frac{1+\sqrt{5}}{2\sqrt{5}}, r_1 = \frac{1-\sqrt{5}}{2\sqrt{5}} \qquad (2.4.3)$$

Будем искать теперь F_k в виде линейной комбинации $r_1^{\,k}, r_2^{\,k}$:

$$F_k = C_1 r_1^{\,k} + C_2 r_2^{\,k} \qquad (2.4.4)$$

Коэффициенты C_1, C_2 определяются из начальных условий (2.4.1) по формулам (2.4.5) :

$$C_1 = \frac{1+\sqrt{5}}{2\sqrt{5}}, C_2 = \frac{1-\sqrt{5}}{2\sqrt{5}} \qquad (2.4.5)$$

Подставляя (2.4.5) в (2.4.4) получим формулу Бине

$$r_1 = \frac{1+\sqrt{5}}{2\sqrt{5}}, r_2 = \frac{1-\sqrt{5}}{2\sqrt{5}} \qquad (2.4.6)$$

Рассмотрим отношения чисел Фибоначчи в виде:

$$\lambda_k = \frac{F_{k+1}}{F_k} \qquad (2.4.7)$$

Переходя к пределу при $k \to \infty$, получим:

$$\lim \lambda_k = \frac{\sqrt{5}-1}{2} = \bar{\lambda} \qquad (2.4.8)$$

Таблица 2.4.1. Числа Фибоначчи F_k и отношения $\lambda_k = \frac{F_{K-1}}{F_k}$

k	**Fk**	**λ_k**	**k**	**Fk**	**λ_k**
1	1	1,0	16	1597	0,618033813400
2	2	0,5	17	2584	0,618034055727
3	3	0,666666666666	18	4181	0,618033963166
4	5	0,6	19	6765	0,618033998521

5	8	0,625	20	10946	0,618033985017
6	13	0,615384615384	21	17711	0,618033990175
7	21	0,619047619047	22	28857	0,618033988205
8	34	0,617647058823	23	46368	0,618033988957
9	55	0,618181818181	24	75025	0,618033988670
10	89	0,617977528089	25	121393	0,618033988780
11	144	0,618055555555	26	196418	0,618033988738
12	233	0,618025751072	27	317811	0,618033988754
13	377	0,618037135278	28	514229	0,618033988748
14	610	0,618032786885	29	832040	0,618033988750
15	987	0,618034447821	30	1346269	0,618033988749

Значение $\bar{\lambda}$ определяет золотое сечение отрезка *[0,1]* : $\frac{1}{\lambda} = \frac{\lambda}{1-\lambda}$,так как является положительным корнем уравнения:

$$\lambda^2 = 1 - \lambda \qquad (2.4.9)$$

Это значение используется в методе *золотого сечения. Золотое сечение* (*золотая пропорция или деление в крайнем и среднем отношении*) — это деление непрерывной величины на две части в таком отношении, при котором меньшая часть так относится к большей части, как большая ко всей величине. Приближенное значение $\bar{\lambda}$ с 12-ю значащими числами будет равно:

$$\bar{\lambda} = 0{,}618033988750\ldots \qquad (2.4.10)$$

Используя (2.4.7) можно показать, что последовательность $\{\lambda_{2k}\}$ является монотонно возрастающей, а последовательность $\{\lambda_{2k+1}\}$ монотонно убывающей, так что при всех k выполняются неравенства $0 < \lambda_{2k} < \bar{\lambda} < \lambda_{2k+1}$.

Зная числа Фибоначчи, можно найти решение уравнений (2.4.1) при любых начальных условиях $F_0 = a$, $F_1 = b$, $F_{k+1}(a,b) = F_{k-1}a + F_k b$.

Отсюда следует, что алгоритм Фибоначчи является непрерывным при любых a и b, но не устойчив.

2.5. Методы нулевого порядка

Рассмотрим задачу минимизации непрерывной функции одной переменной $\varphi(x)$ на промежутке $[a,b]$:

$$\min\{ \varphi(x) \,/\, a \leq x \leq b \} \tag{2.5.1}$$

Для решения этой задачи разработаны специальные методы поиска с однократным вычислением значения функции на каждом шаге, не использующие производные. К ним относятся: метод золотого сечения, метод оптимального поиска, метод симметричного поиска и другие.

Все эти методы характеризуются одними и теми же параметрическими свойствами алгоритмов и ограничиваются классом унимодальных функций. В п. 2.5.1. введено понятие унимодальной функции и рассмотрены её свойства.

2.5.1. Унимодальные функции.

Рассмотрим класс унимодальных функций и их свойства.

Определение 2.5.1. *Функция $\varphi(x)$ называется унимодальной на промежутке $[a, b]$, если она:*

1. *строго убывает на отрезке $[a,x_1^*]$;*
2. *постоянна на трезке $[x_1^*, x_2^*]$;* (2.5.2)
3. *строго возрастает на отрезке $[x_2^*, b]$.*

Если при этом $x_1^*= x_2^*$,то функция $\varphi(x)$ называется строго унимодальный. Точки x_1^* и x_2^*, могут совпадать с концевыми точками a и b.

Примером унимодальной функции на промежутке $[a, b]$, является любая выпуклая функция на этом промежутке, локализующем экстремальную точку (max или min).

Определение 2.5.2 Точку $x' \in [a,b]$, в которой вычисляется значение z'= (x'), назовем *пробной точкой*, а пару чисел (x', z') -.*пробой.*

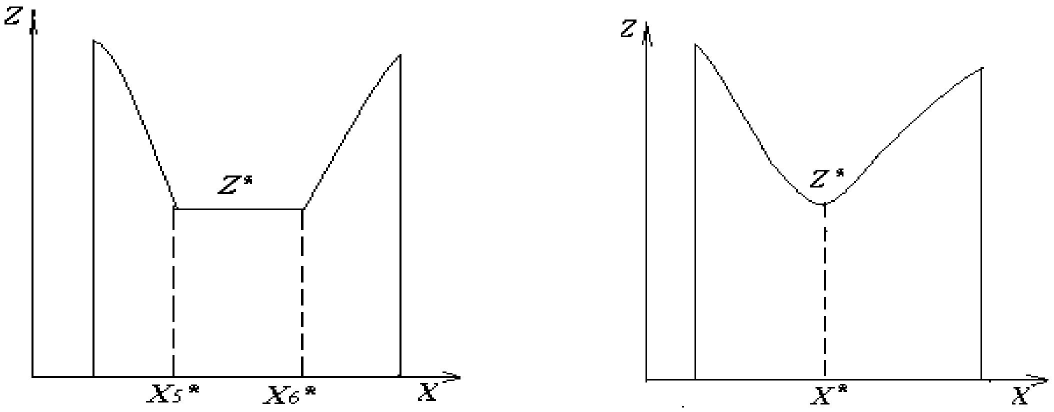

Рис. 2.5.1. Унимодальная и строго унимодальная функции.

Определение 2.5.3. *Любой промежуток [a',b'] $\subset$ [a,b], содержащий хотя бы одну точку x*$\in$ X*, назовем промежутком локализации точки минимума x*, а промежуток [c',d'], содержащий z* - промежутком локализации минимального значения z*.*

Унимодальные функции обладают замечательными свойствами, на которых основаны все методы поиска их экстремума. Эти свойства отражает следующая лемма.

Лемма 2.5.1. *Любые две пробные точки x,y$\in$ (a,b), x<y, позволяют сузить промежуток локализации x* до [a',b'] согласно правилам:*

$$
\begin{aligned}
&1)\ a' = a,\ b'=y,\ \text{если}\ \varphi(x) < \varphi(y);\\
&2)\ a'=x, b'=b, \text{если}\ \varphi(x)>\varphi(y); \qquad (2.5.3)\\
&3)\ a' = x,\ b'=y,\ \text{если}\ \varphi(x) = \varphi(y).
\end{aligned}
$$

<u>*Доказательство:*</u>

Пусть имеет место первый случай и ни одна точка *x* не принадлежит отрезку[a',b'], x*$\notin$[a',b']* (рис. 2.5.2)

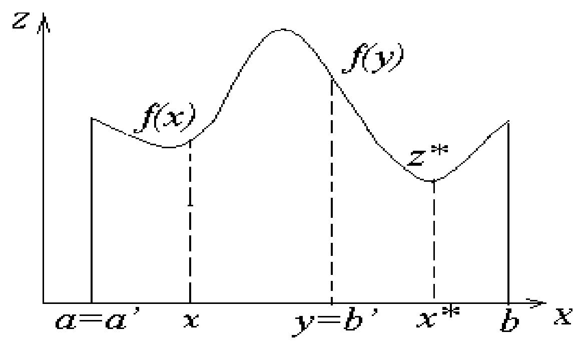

Рис.2.5.2. Вид функции $f(x)$ при $x^* \notin [a',b']$.

Тогда функция не будет монотонно убывающей в промежутке $[a, x_1^*]$, что противоречит свойству унимодальности функции $\varphi(x)$. Значит предположение не верно, и, действительно, $x^* \in [a',b']$.

Второй случай рассматривается аналогично. В третьем случае, в силу условий *(2.5.2)* точки x *и* y не могут одновременно принадлежать $[a, x_1^*]$ и $[x_2^*,b]$. Поэтому промежуток $[x, y]$ обязательно содержит хотя бы одну точку $x^* \in X^*$.

Замечание: В методах поиска третье правило из *(2.5.3)* обычно не используется, а случай $\varphi(x) = \varphi(y)$ присоединяется к первому правилу.

2.5.2. Симметричный поиск. Метод симметричного поиска.

Согласно правилу *(2.5.3)*, любой из промежутков $[a, y]$ *и* $[x, b]$ в классе унимодальных функций с равной вероятностью может оказаться промежутком локализации x^*. Поэтому целесообразно выбирать пробные точки таким образом, чтобы эти промежутки имели одинаковую длину.

Отсюда следует, что точки x_k и y_k на всех шагах процесса должны располагаться симметрично относительно концов промежутка $[a_k,b_k]$ (рис.2.5.3). При этом на $k+1$-ом шаге в промежутке $[a_k,b_k]$ уже содержится пробная точка от k-го шага, так что требуется лишь одна пробная точка, симметричная ей.

Все алгоритмы такого рода можно объединить в одно семейство, вводя параметр λ, который определеяет положение первых двух пробных точек в промежутке $[a, b]$.

Схема разбиения интервала $[a, b]$, содержащую оптимальную точку представлена на рис.2.5.3.

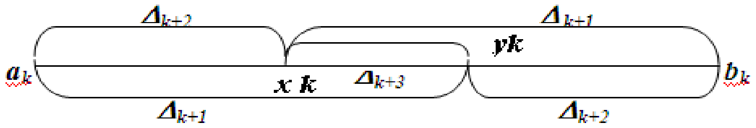

Рис. 2.5.3. Схема разбиения $[a_k,b_k]$.

Алгоритм симметричного поиска.

Задаем значения a, b, λ $(\lambda\in(0,5;\ 1))$, $\varepsilon>0$.

Шаг 1. Полагаем $a_0= a$, $b_0= b$, $\Delta_0= b_0- a_0$, $\Delta_1=\lambda*\Delta_0$, $\Delta_2=\Delta_0-\Delta_1$, $x_0= a_0+\Delta_2$, $y_0= b_0-\Delta_2$. Вычисляем $\varphi(x_0)$, $\varphi(y_9)$ и полагаем $k=1$.

Шаг 2. Вычисляем $\Delta_{k+2}=\Delta_k-\Delta_{k+1}$ и проверяем: если $\Delta_{k+2}\leq 0$, то алгоритм прекращает работу, иначе переходим к *шагу 3*.

Шаг 3. *a)* Если $\varphi(x_{k-1})\leq\varphi(y_{k-1})$, то полагаем $a_k= a_{k-1}$,$b_k= y_{k-1}$, $y_k= x_{k-1}$,$\varphi(y_k) = \varphi(x_{k-1})$. Находим $x_k= a_k+\Delta_{k+2}$, вычисляем $\varphi(x_k)$ и переходим к ***Шагу 4***.

b) Если $\varphi(x_{k-1}) > \varphi(y_{k-1})$, то полагаем $a_k= x_{k-1}$, $b_k= b_{k-1}$, $x_k= y_{k-1}$, , $\varphi(x_k)= \varphi(y_{k-1})$. Находим $y_k= b_k-\Delta_{k+2}$, вычисляем $\varphi(y_k)$ и переходим к ***Шагу 4***.

Шаг 4. Если $\Delta_k>\varepsilon$, то полагаем $k=k+1$ и переходим к ***Шагу 2***. Если же $\Delta_k\leq\varepsilon$, то процесс окончен, переходим к ***Шагу 5***.

Шаг 5. Выписываем ответ a_k,, b_k, y_k,, $\varphi(y_k)$ или a_k,, b_k , x_k, $\varphi(x_k)$, если окончание процесса происходит после ***Шага 3, пункт а)*** или если окончание процесса происходит после ***Шага 3, пункт b).***

Предложенный алгоритм симметричного поиска вырабатывает систему вложенных промежутков$\{[a_k,, b_k]\}$, пробных точек $\{x_k,y_k\}$ и значений $\{\varphi(x_k).\varphi(y_k)\}$. Для иллюстрации работы алгоритма рассмотрим простейший пример.

Пример 2.5.1. Найти минимум функции $\varphi(x)=/x\text{-}0{,}3/$ на отрезке $[0, 1]$, то есть решить задачу: $min\ \{\ |x-0{,}3|\ /\ 0\leq x\leq 1\ \}$

Возьмем значение $\lambda= 0,62$, $\Delta_0=1$. Тогда $\Delta_1=\lambda\Delta_0=0,62$, $\Delta_2=\Delta_0-\Delta_1=0,38$. Зададим $\varepsilon=0,01$.

Результаты решения примера по алгоритму симметричного поиска представлены в таблице *2.5.1*.

Таблица 2.5.1. Результаты решения примера 2.5.1.

k	Δ_{k+2}	a_k	b_k	x_k	y_k	$\varphi(x_k)$	$\varphi(y_k)$
0	0,38	0	1	0,38	0,62	0,08	0,32
1	0,24	0	0,62	0,254	0,38	0,06	0,08
2	0,14	0	0,38	0,14	0,24	0,16	0,06
3	0,10	0,14	0,38	0,24	0,28	0,06	0,02
4	0,04	0,24	0,38	0,28	0,34	0,02	0,04
5	0,06	024	0,34	0,30	0,28	0,00	0,02
6	-0,02						

Из ***таблицы 2.5.1***. видно, что алгоритм прекращает работу по условию **Шага 2** после 5-ой итерации, хотя требуемая точность еще не достигнута. Это вызвано тем, что на 5-ой итерации точка x_5 легла правее y_5, что привело к отрицательности $\Delta_8=-0,02$ и к нарушению, в дальнейшем, вложенности промежутков.

2.5.3. Исследование алгоритма симметричного поиска

Согласно ***Шагу 2*** алгоритм симметричного поиска реализует *k-ю* итерацию лишь при
условии, что

$$\Delta_{k+2}(\lambda) > 0, \qquad (2.5.4)$$

иначе он прекращает работу. Условие *(2.5.4)* обеспечивает невырожденность и вложенность промежутков $[a_k, b_k]$, но в то же время делает невозможным достижение требуемой точности решения задачи.

Докажем теорему о необходимом и достаточном условии выработки алгоритмом симметричного поиска системы вложенных промежутков.

Теорема 2.5.1. (*о необходимом и достаточном условии* выработки алгоритмом симметричного поиска системы вложенных промежутков)

Для того, чтобы алгоритм симметричного поиска при $\varepsilon=0$ вырабатывал n вложенных промежутков $[a_k, b_k] \subset [a_{k-1}, b_{k-1}]$, $k=1,2,\ldots,n$, локализующих точку x^*, необходимо и достаточно, чтобы значение параметра λ удовлетворяло условиям:

$$\lambda_k<\lambda<\lambda_{k+1}, \text{ если } k\text{-четное},$$

$$\lambda_{k+1}<\lambda<\lambda_k, \text{ если } k\text{- не четное}, \qquad (2.5.5)$$

где $\lambda_k=\frac{F_{k-1}}{F_k}$ *– отношение чисел Фибоначчи.*

<u>*Доказательство*</u>

Из алгоритма поиска следует, что при $k\geq 1$ выполняются соотношения:

$$b_{k-1}- a_{k-1}=\Delta_{k-1};\; b_{k-1}- x_{k-1}= y_{k-1} - a_{k-1}=\Delta_k;$$

$$x_{k-1} - a_{k-1}= b_{k-1}- y_{k-1}=\Delta_{k+1};\; y_{k-1} - x_{k-1}=\Delta_{k+2} \qquad (2.5.6)$$

Но, в силу условия *(2.5.4)*, все $\Delta_i>0$, $i=0,1,2,\ldots,k+2$, поэтому из (2.5.6) следует, что

$$a_{k-1} < x_{k-1} < y_{k-1} < b_{k-1}\,. \qquad (2.5.7)$$

Это обеспечивает реализуемость k-го шага, т.е. невырожденность промежутка $[a_k, b_k]$ и его вложенность в промежуток $[a_{k-1}, b_{k-1}]$. С другой стороны условия (2.5.7) равносильны условию *(2.5.8)*

$$0 \leq \Delta_{k+1}(\lambda) < \Delta_k(\lambda) \qquad (2.5.8)$$

Используя формулы (2.4.12) и метод математической индукции найдем зависимость Δ_k от λ

$$\Delta_k(\lambda) =(-1)^{k-1}[F_{k-1}\lambda- F_{k-2}]\,\Delta_0\,, \qquad (2.5.9)$$

где F_k – числа Фибоначчи.

Тогда, разрешив неравенства *(2.5.8)* относительно λ, получим:

$$\lambda_k<\lambda<\lambda_{k+1}, \text{ если } k\text{-четное},$$

$\lambda_{k+1}<\lambda<\lambda_k$, *если k- не четное.* (2.5.10)

Если алгоритм симметричного поиска вырабатывает n вложенных помежутков, то условия (2.5.10) должны выполняться при всех значениях $k=1,2,\ldots,n$. Однако $\lambda_{2k} \nearrow \bar{\lambda}$, а $\lambda_{2k+1} \searrow \bar{\lambda}$, поэтому выполнение условий *(2.5.5)* при $k=n$ обеспечивает выполнение условий *(2.5.10)* при всех $k \le n$, так что справедливо утверждение теоремы *2.5.1* о необходимом и достаточном условии выработки алгоритмом симметричного поиска системы вложенных промежутков.

Промежуток допустимых значений λ с ростом n ,быстро сужается. Например, используя таблицу 2.4.1, находим, что

$$\lambda_9 - \lambda_{10} < 2 * 10^{-4}\ ; \qquad \lambda_{19} - \lambda_{20} < 2 * 10^{-8}\ ; \qquad \lambda_{29} - \lambda_{30} < 2 * 10^{-12}$$

Следствие 2.5.1. Алгоритм симметоричного поиска вырабатывает бесконечную последовательность вложенных промежутков $\{[\ a_k,\ b_k]\}$ лишь в единственном случае, когда $\lambda=\bar{\lambda}$, т.е. когда алгоритм реализует метод золотого сечения.

В методе золотого сечения все промежутки разбиваются в одном и том же отношении, определяемом $\bar{\lambda}$, откуда следует, что метод золотого сечения сходится, так как

$$b_k - a_k = \Delta_k(\bar{\lambda}) = \bar{\lambda}^k \Delta_0 \to 0.$$

В работах [1], [9] показано, что, если снять ограничение *(2.5.4)* на шаге 2 и условие окончания процесса по заданной точности ε на шаге 4, то алгоритм станет неустойчивым, будет формально вырабатывать бесконечные последовательности чисел $\{a_k, b_k\}, \{x_k, y_k\}, \{\varphi(x_k), \varphi(y_k)\}$ и, следовательно, не будет сходиться.

Теорема 2.5.2. (*об устойчивости алгоритма симметричного поиска*)
Если снять ограничение (2.5.4) на шаге 2 и условие окончания процесса по заданной точности ε *на шаге 4, то алгоритм симметричного поиска станет неустойчивым.*

<u>*Доказательство*</u>

Обозначим через $A(\lambda)$ – алгоритм симметричного поиска. Алгоритм $A(\lambda)$ представим в виде композиции вспомогательных алгоритмов $B(\lambda)$ *и* $C(\lambda)$: $A(\lambda)=B(\lambda)\ x\ C(\lambda)$.

Причем алгоритм $C(\lambda)$ вырабатывает последовательность $\{\Delta_k(\lambda)\}$ по рекуррентным формулам (2.5.11):

$$\Delta_1=\lambda\times\Delta_0,,\quad \Delta_{k+1}=\Delta_{k-1} - \Delta_k\ ,\ k\geq 1\ , \tag{2.5.11}$$

а алгоритм $B(\lambda)$ вырабатывает последовательности $\{a_k,, b_k\}$, $\{y_k, \varphi(y_k)\}$, $\{x_k, \varphi(x_k)\}$ согласно правилам шага 3 с использованием $\Delta_{k+1}(\lambda)$.

Применяя метод математической индукции, найдем зависимость Δ_k от λ:

$$\Delta_k(\lambda) =(-1)^{k-1}(F_{k-1}\lambda - F_{k-2})\ \Delta_0\ , \tag{2.5.12}$$

где F_k – числа Фибоначчи. Откуда видно, что все $\Delta_k(\lambda)$ линейно зависят от λ и, следовательно, непрерывны по λ. Это означает непрерывность алгоритма $C(\lambda)$. Используя формулу (2.5.12), можно записать:

$$/\Delta_k(\lambda) - \Delta_k(\lambda_0)/ = F_{k-1}\Delta_0/\lambda-\lambda_0/\to\infty$$

при $k\to\infty$ для всех $\lambda \neq \lambda_0$, так как $F_{k-1}\to\infty$.

В силу произвольности λ_0 следует, что алгоритм $C(\lambda)$ неустойчив при любом $\lambda\in R_1$. Следовательно, будет неустойчив и алгоритм $A(\lambda)$, включающий $C(\lambda)$ как составную часть. Теорема доказана.

Следствие 2.5.2. *Метод золотого сечения, реализуемый алгоритмом* $A(\lambda)$ *при* $\lambda=\bar{\lambda}$ *является неустойчивым и, следовательно, практически не сходится.*

При численной реализации алгоритма на ЭВМ , в силу своей неустойчивости, алгоритм прекращает работу гораздо раньше, чем это предусмотрено условиями алгоритма. Метод золотого сечения становится конечным, так как $\bar{\lambda}$ - иррационально и не может быть задано точно, а также вследствие влияния погрешностей вычисления. Метод Фибоначчи довольно быстро теряет свойство оптимальности, так что его преимущество чисто теоретическое.

Поэтому в практических расчетах всегда целесообразно выбирать λ, которое аппроксимирует $\bar{\lambda}$ с наибольшей допустимой точностью.

В работах [9], [10] проведено экспериментальное исследование алгоритма симметричного поиска, которое показало, что существует эмпирическая зависимость между числом знаков после запятой (точностью решения задачи) и числом вложенных промежутков $\{a_k,,b_k\}$,вырабатываемых алгоритмом.

2.5.4. Алгоритм симметричного поиска с восстановлением

При решении конкретных задач часто возникают проблемы, при которых алгоритм симметричного поиска становится вырожденным и после ряда шагов начинает расходиться, т.е. величина $\Delta_k= b_k - a_k$ вместо убывания начинает принимать различные по знаку и возрастающие по величине значения.

В работе [9] показано, что для блокировки вырожденности алгоритма симметричного поиска систематически производится восстановление алгоритма, которое заключается в следующем. Принимая на некоторой (например, *k-ой*) итерации промежуток $[a_k, b_k]$ за начальное приближение $[a_0, b_0]$, продолжаем процесс с ***Шага 1***, т.е. с начала. Это восстановление осуществляется автоматически. Для этого к алгоритму симметричного поиска добавляется ещё один шаг - ***Шаг 5***.

Шаг 5. Если $\Delta_k < \alpha*\Delta_{k-1}$, то полагаем $a_0= a_k$, $b_0= b_k$ и переходим к ***Шагу 1***. Иначе переходим к ***Шагу 2***.

После задания a, от ***Шага 4*** по условию $\Delta_k > \varepsilon$ производится переход к ***Шагу5***. Если $\varepsilon=0$,то алгоритм симметричного поиска с восстановлением, при любых λ и $0,5<\alpha<\lambda<1$ вырабатывает бесконечную последовательность вложенных промежутков, стягивающихся к точке минимума x^*.

Алгоритм симметричного поиска с восстановлением сходимости является асимптотически устойчивым при всех значениях λ и α. Доказательство этого утверждения предложено в работе Тетерева А.Г «Анализ сходимости и устойчивости методов одномерной оптимизации» [9]. Параметр α определяет допуск на изменение λ в процессе счета.

В случае, когда $\lambda=\bar{\lambda}$, выполняется соотношение $\Delta_{k+1}(\bar{\lambda})=\bar{\lambda}\times\Delta_k(\bar{\lambda}),\ k=0,1,2,\dots$

.

Если же $\lambda\neq\bar{\lambda}$, то соотношение между $\Delta_{k+1}(\lambda)$ и $\Delta_k(\lambda)$ на каждом шаге изменяется по формуле: $\Delta_{k+1}(\lambda)=\tilde{\lambda}_k\times\Delta_k(\lambda)$, причем $\tilde{\lambda}_k$ колеблется около $\bar{\lambda}$.

Исследования показали, что как только $\tilde{\lambda}_k$ будет меньше α, то, согласно алгоритму симметричного поиска происходит восстановление сходимости. Поэтому с помощью α можно полностью регулировать эффективность процесса.

Практика показала, что если выбрать α близким к 0,5 сверху, то восстановление срабатывает лишь в самый последний момент, когда алгоритм должен был прекратить работу по условию $\Delta_{k+2}\leq 0$ ***Шага 2*** алгоритма.

Если же выбрать α очень близким к λ,, то восстановление будет происходить слишком часто, что замедляет сходимость процесса.

2.5.5. Несимметричный поиск. Метод несимметричного поиска

Исследование алгоритма симметричного поиска показало, что он является неустойчивым при всех значениях λ и α. Поэтому возникла необходимость разработки более устойчивого алгоритма, который получил название: алгоритм несимметричного поиска [1]. Будем вновь использовать процесс формирования пробных точек x_k, y_k *(см. п. 2.5.2)* и правило сужения промежутка локализации экстремума *(*2.5.3*)*. *С*огласно формулам (2.5.6), между пробными точками выполняется соотношение y_{k-1} - x_k.Возьмем это соотношение за основу. Тогда нет необходимости использовать концевые точки промежутка $[\ a_k,,\ b_k]$ в явном виде. Изменим назначение точек x_k, y_k, таким образом, что y_k,- играет роль пробной точки, а x_k –роль приближения.

Формирование последовательности $\{\Delta_k\ (x)\}$ будем выполнять с помощью рекуррентных соотношений

$$\Delta_k = \lambda^*\Delta_{k-1},\ \text{где}\ \lambda \in R_1,\quad \Delta_0\ \text{задано},\ \Delta_0 \# 0 \qquad (2.5.11)$$

Процесс поиска начинаем с левой точки промежутка $[a, b]$. Алгоритм несимметричного поиска ориентирован на его применение в методах многомерной оптимизации.

2.3.2

Задаем концевые точки промежутка a, b, параметр λ $(\lambda \in (0; 1))$,погрешность $\varepsilon > 0$, число M –такое,что $M > \varphi(x)$ на отрезке $[a, b]$.

Шаг 1. Вычисляем $\Delta_0 = b - a$, $\Delta_1 = \lambda^*\Delta_0$, $\Delta_2 = \lambda\Delta_1 = \lambda^2{}^*\Delta_0$, $x_0 = a$, ,. $\varphi_0 = M$, полагаем $k=1$.

Шаг 2. Вычисляем $\Delta_{k+2} = \lambda\Delta_{k+1}$ и $\varphi(y_{k-1})$.

Шаг 3. *а)* Если $\varphi(x_{k-1}) \le \varphi(y_{k-1})$ *(см. рис.2.5)* , то полагаем $x_k = x_{k-1}$, $\varphi(x_k) = \varphi(x_{k-1})$.

Находим: $y_k = x_k - \Delta_{k+2}$, *если* $x_{k-1} < y_{k-1}$; $y_k = x_{k+} + \Delta_{k+2}$, *если* $x_{k-1} > y_{k-1}$ и переходим к **Шагу 4**.

b) Если $\varphi(x_{k-1}) > \varphi(y_{k-1})$ *(см. рис.2.6)* , то полагаем $x_k = y_{k-1}$, , $\varphi(x_k) = \varphi(y_{k-1})$.

Находим: $y_k = x_k + \Delta_{k+2}$, *если* $x_{k-1} < y_{k-1}$; $y_k = x_k - \Delta_{k+2}$, *если* $x_{k-1} > y_{k-1}$ и переходим к **Шагу 4.**

Шаг 4. Если $\Delta_{k+2} \le \varepsilon\Delta_0$, то процесс оконечен. Экстремальная точка $(x_k, \varphi(x_k))$. Иначе полагаем $k=k+1$ и переходим к **Шагу 2.**

При значении $\lambda = \bar{\lambda}$ алгоритм несимметричного поиска эквивалентен алгоритму симметричного поиска, так как он вырабатывает, по существу, те же пробные точки x_k , y_k, только в иных обозначениях, и неявно порождает такие же промежутки $[a_k, b_k]$, причем номера пробных точек на одну единицу выше номеров промежутков:

$a_{k-1} = x_k - \Delta_{k+1}$, *если* $x_k < y_k$; $\qquad b_{k-1} = x_k + \Delta_{k+1}$, *если* $x_k < y_k$

$a_{k-1} = y_k - \Delta_{k+1}$, *если* $x_k > y_k$ $\qquad b_{k-1} = y_k + \Delta_{k+1}$, *если* $x_k > y_k$

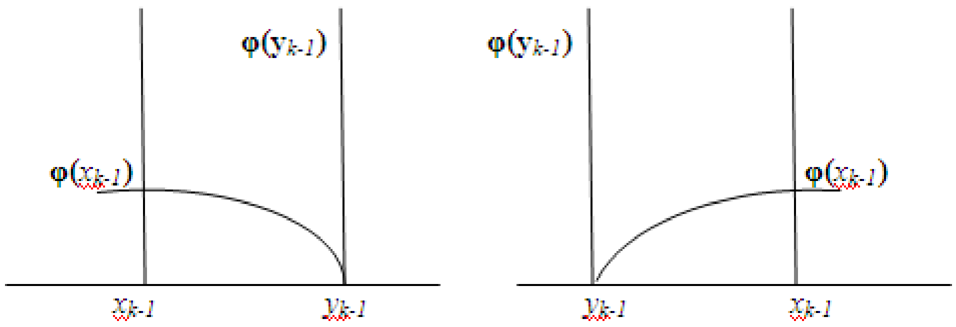

Рис. 2.5.4. Выбор промежутка локализации x^*, случай $\varphi(x_{k-1}) \le \varphi(y_{k-1})$.

Следовательно, алгоритм несимметричного поиска также реализует метод золотого сечения.

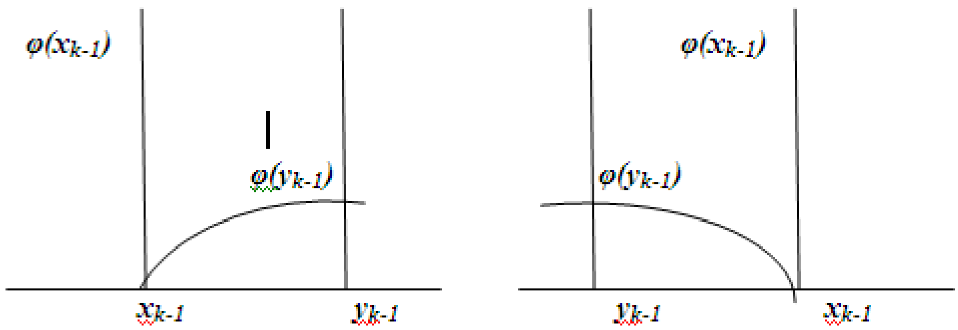

Рис. 2.5.5. Выбор промежутка локализации x^*, случай $\varphi(x_{k-1}) > \varphi(y_{k-1})$.

В работе [1] проведен всесторонний анализ алгоритма несимметричного поиска, который показал, что указанный алгоритм устойчив и сходится за конечное число шагов. Полная погрешность решения не более чем на порядок выше величины $\varepsilon\Delta_0$.

Таким образом, погрешность алгоритма несимметричного поиска вполне контролируема, и, выбирая подходящим образом ε , Δ *и* λ, можно определить решение задачи (2.21) с любой наперед заданной точностью.

2.5.6. Определение начального промежутка

Обычно методы одномерного поиска применяются для определения длины шага в методах многомерной оптимизации и начальный промежуток $[a_0, b_0]$. как правило, неизвестен. Для его определения можно использовать алгоритм несимметричного поиска, применяя его в обратном направлении.

Рассмотрим поиск начального промежутка $[a_0,b_0]$ на луче $[a,\infty]$, используя алгоритм метода несимметричного поиска в обратном направлении

2.5.7. Алгоритм поиска начального промежутка.

Задаем $a, \Delta_0, \mu=1/\bar{\lambda}$, $\varepsilon>0$, число M, $M>\varphi(x)$ на отрезке $[a, \infty]$.

Шаг 1. Полагаем $x_0=a; M=\varphi(x_0)$, $y_0=x_0+\Delta_0; k=1$.

Шаг 2. Вычисляем $\varphi(y_{k-1})$. Если $\varphi(x_{k-1}) > \varphi(y_{k-1})$, то полагаем $x_k=y_{k-1}$, , $\varphi(x_k)=\varphi(y_{k-1})$, $\Delta_k(\mu)=\mu^*\text{-}\Delta_{k-1}(\mu)$, $y_k=x_k+\Delta_k(\mu)$, $k=k+1$ и повторяем *шаг2*, иначе переходим к *шагу3*.

Шаг 3. Полагаем $\Delta_{k+1}(\lambda)=\lambda^2{*}\Delta_{k-1}(\mu), y_{k-1}=x_{k-1}+\Delta_{k+1}(\lambda)$ и переходим к шагу 4 (или *шагу 2 алгоритма несимметричного поиска.*)

Шаг 4. Вычисляем $\Delta_{k+2}=\lambda\Delta_{k+1}$ и $\varphi(y_{k-1})$. Возможны 2 случая:

a) Если $\varphi(x_{k-1}) \le \varphi(y_{k-1})$, то полагаем $x_k=x_{k-1}$, $\varphi(x_k)=\varphi(x_{k-1})$.

Находим: $y_k = x_k - \Delta_{k+2}$, *если* $x_{k-1} < y_{k-1}$; $y_k = x_k + \Delta_{k+2}$, *если* $x_{k-1} > y_{k-1}$.

b) Если $\varphi(x_{k-1}) > \varphi(y_{k-1})$, то полагаем $x_k=y_{k-1}$, , $\varphi(x_k)=\varphi(y_{k-1})$.

Находим: $y_k = x_k + \Delta_{k+2}$, *если* $x_{k-1} < y_{k-1}$; $y_k = x_k - \Delta_{k+2}$, *если* $x_{k-1} > y_{k-1}$

.

В результате работы алгоритма происходит наращивание числа промежутков $[a_k,b_k]$ до тех пор, пока не выполнятся условия

$$\varphi(x_{k-2}) > \varphi(x_{k-1}) \le \varphi(y_{k-1}).$$

Для унимодальной функции это означает, что $x^* \in [x_{k-2}, y_{k-1}]$. отрезок $[x_{k-2}, y_{k-1}]$ и принимается в качестве начального промежутка при переходе к алгоритму несимметричного поиска. Откуда

$$\Delta_0 = y_{k-1} - x_{k-2} = \Delta_{k-2}(\mu) + \Delta_{k-1}(\mu) = \Delta_k(\mu),$$

так как $\mu^2 = 1 + \mu$. Далее полагаем $x_0 = x_{k-2}$, тогда согласно алгоритму несимметричного поиска имеем:

$$x_1 = x_0 + \lambda^2 * \Delta_0 = x_{k-2} + \lambda^2 \Delta_k(\mu) = x_{k-2}(\mu) = x_{k-1}(\mu).$$

Следовательно, момент перехода к алгоритму несимметричного поиска соответствует $k=2$, и необходимо вычислить пробную точку:

$$y_1 = x_1 + \Delta_3(\lambda) = x_1 + \lambda^3 \Delta_0 = x_1 + \lambda^3 \Delta_k(\mu) = x_1 + \lambda^2 \Delta_{k-1}(\mu).$$

Этим объясняется условие перехода к алгоритму несимметричного поиска на шаге 3. \

2.6. Методы первого порядка

Рассмотрим методы оптимизации, в которых используются первые производные целевой функции. К ним относятся метод Больцано, метод касательных и другие методы [3]. В общем случае они не имеют особых преимуществ перед методами поиска и даже уступают им по эффективности, так как вычисление каждого значения производной равносильно вычислению двух значений функции.

Преимущество методов первого порядка проявляется лишь при наличии простых формул вычисления производных. Этот фактор не играет большой роли при выборе метода.

Метод Больцано (или метод деления отрезка пополам) привлекает своей простотой и надежностью реализации. В свою очередь, метод касательных служит прототипом метода глобального поиска, который рассмотрен в п. 2.8.

2.6.1. Выпуклые функции

Методы первого порядка применяются, как правило, для оптимизации выпуклых дифференцируемых функций [1, 3].

Определение 2.6.1. *Функция $\varphi(x)$, $x\in[a,b]$, называется выпуклой, если при любых x', $x''\in[a,b]$, и всех α, $0\le\alpha\le 1$ выполняется неравенство*

$$\varphi((1-\alpha)x'+\alpha x'')\le(1-\alpha)\varphi(x')+\alpha\varphi(x'') \qquad (2.6.1)$$

и строго выпуклой, если это неравенство выполняется строго для всех $0<\alpha<1$.

При изменении α в пределах $[0,1]$ точка $x(\alpha)=(1-\alpha)x'+\alpha x''$ пробегает от x' до x'' весь отрезок $[x', x'']$, а значение $z(\alpha)=(1-\alpha)z'+\alpha z''$ – все значения от $z'=\varphi(x')$ до $z''=\varphi(x'')$, лежащие на хорде z',z''.(см. рис.2.6.1.)

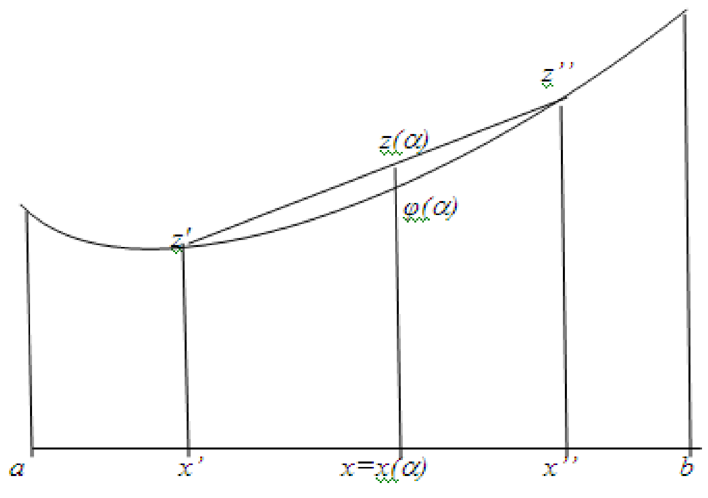

Рис. 2.6.1. Геометрическая интерпретация выпуклой функции

Поэтому условие *(2.6.1)* геометрически означает, что на любом отрезке $[x', x'']\subset[a,b]$ график выпуклой функции $\varphi(x)$ лежит не выше хорды, соединяющей концевые точки (x',z') и (x'',z''). Примерами выпуклых функций на произвольном отрезке $[a, b]$ могут служить функции: $\varphi(x)=x$; $\varphi(x)=|x-x_0|$; $\varphi(x)=ax^2+bx+c$.

Определение 2.6.2. *Функция $\psi(x)$, $x\in[a, b]$, называется вогнутой (строго вогнутой), если функция $\varphi(x)=-\psi(x)$, $x\in[a, b]$, выпукла (строго выпукла).*

Выпуклые функции обладают рядом замечательных свойств. Они являются непрерывными во всех внутренних точках $x \in (a, b)$ и имеют левую и правую производные в этих точках.

Кроме того, если существуют ограниченные производные $\varphi'(a+0)$ и $\varphi'(b-0)$, то выпуклая функция $\varphi(x)$ удовлетворяет условию Липшица с константой L:

$$L = \max \{ |\varphi'(a+0)|, |\varphi'(b-0)| \}, \qquad (2.6.2)$$

Рассмотрим некоторые свойства выпуклых функций, представленные теоремой 2.6.1.

Теорема 2.6.1. *Для дифференцируемой функции $\varphi(x)$, $x \in [a, b]$, следующие условия эквивалентны:*

1. *(x) выпуклая на $[a, b]$, т.е. выполняется условие*

$$\varphi(x) \geq \varphi'(x_0)(x - x_0) + \varphi(x_0), \; x, x_0 \in [a, b] \qquad (2.6.3)$$

2. *$\varphi'(x)$ не убывает на $[a, b]$* (2.6.4)

3. *$\varphi''(x) \geq 0$ на отрезке $[a, b]$* (2.6.5)

<u>Доказательство.</u>

Пусть $\varphi(x)$ выпукла. Тогда из условия (2.6.3) при $x' = x_0$ и $x'' = x$ имеем:

$$\varphi(x_0 + \alpha(x - x_0)) \leq \varphi(x_0) + \alpha[\varphi(x) - \varphi(x_0)] \qquad (2.6.6)$$

Представим (2.6.6) в виде: $\varphi(x) \geq \varphi(x_0) + 1/\alpha\{\varphi(x_0 + \alpha(x - x_0)] - \varphi(x_0)\}$.

Перейдем к пределу при $\alpha \to 0$, в результате получим условие (2.6.3). Далее из условия (2.6.3) получим (2.6.4). Поменяв ролями точки x и x_0 в условии (2.6.3) , получим:

$$\varphi(x_0) \geq \varphi(x) + \varphi'(x)(x_0 - x), \quad x, x_0 \in [a, b]$$

Сопоставляя это неравенство с (2.6.3), можно записать, что

$$\varphi'(x)(x - x_0) \geq \varphi'(x_0)(x - x_0) \qquad (2.6.7)$$

Отсюда следует, что производная $\varphi'(x)$ является неубывающей функцией на отрезке $[a,b]$, т.е. выполняется условие (2.6.4).

Покажем теперь, что из условия (2.6.4) следует условие (2.6.3)

$$\varphi((1-\alpha)x'+\alpha x'') - [(1-\alpha)\varphi(x')+\alpha\varphi(x'')] = [\varphi(x'+\alpha(x'-x'')) - \varphi(x')] - \alpha[\varphi(x'')-$$

$$\varphi(x')] = \alpha(x''-x')\int_a^b [\varphi'(x'+\alpha\tau(x'-x'')) - \varphi'(x'+\tau(x'-x''))]\,\partial\tau \le 0,$$

так как подинтегральная функция неположительная вследствие монотонности $\varphi'(x)$.

Таким образом, условия (2.6.3) и (2.6.4) эквивалентны. Теорема полностью доказана.

Условие (2.6.3) имет простую геометрическую интерпретацию, представленную на рис.2.6.2.

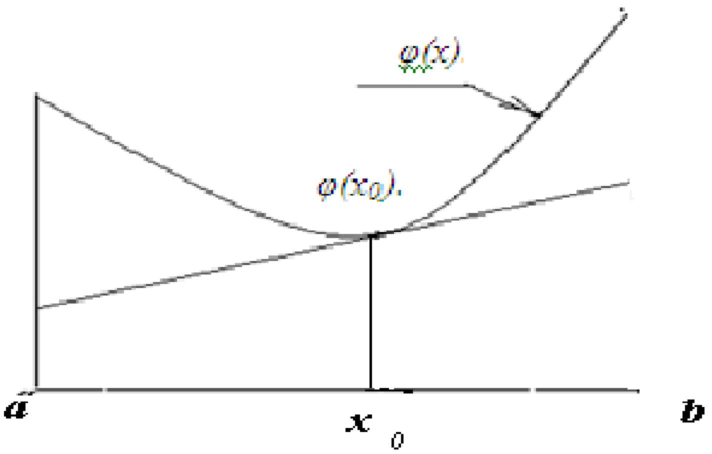

Рис.2.6.2. Геометрическая интерпретация условия (2.6.3).

Выражение в правой части этого условия

$$\varphi'(x_0)(x-x_0)+\varphi(x_0) \qquad (2.6.8)$$

представляет собой уравнение касательной к графику функции $\varphi(x)$ в точке x_0. Поэтому неравенство (2.6.3) означает, что график функции $\varphi(x)$ лежит над касательной.

2.6.2.Критерии оптимальности первого порядка

Рассмотрим экстремальные свойства выпуклых функций.

Теорема 2.6.2. Любая выпуклая функция $\varphi(x)$, $x\in[a, b]$, *является унимодальной и в случае дифференцируемости* её множество оптимальности $X^*\in[a^*, b^*]$ *определяется условием:*

$$X^*=\{ x^*\in[a, b]/ \varphi'(x^*)(x-x^*)\geq 0,\ \forall x^*\in[a, b]\} \qquad (2.6.9)$$

<u>*Доказательство*</u>

Для унимодальности функции $\varphi(x)$ необходимо показать, что:

1) любой локальный экстремум функции $\varphi(x)$ на отрезке $[a, b]$ является и глобальным экстремумом;
2) множество оптимальности X^* функции $\varphi(x)$, $x\in[a, b]$ – выпукло.

Допустим, что точки x_1,x_2 –точки локального минимума функции $\varphi(x)$, $x\in[a, b]$ *и* $\varphi(x_1)\leq \varphi(x_2)$, Тогда в силу выпуклости $\varphi(x)$

$$\varphi(x(\alpha)),\leq(1-\alpha)\varphi(x_1)+\alpha\varphi(x_2) \leq \varphi(x_1),\quad \forall\alpha\in[0,1], \qquad (2.6.10)$$

где $x(\alpha)=(1-\alpha)x_1+\alpha x_2$.

При $\alpha \to 1$ точка $x(\alpha) \to x_2$. Значит, в любой окрестности $O(x_2)$ имеются такие точки $x=x(\alpha)$, что $\varphi(x) < \varphi(x_2)$. Это противоречит условию локального минимума в точке x_2. Следовательно, наше предположение не верно, и тогда $\varphi(x_1)=\varphi(x_2)$. Тем самым первое утверждение доказано.

Пусть $x_1, x_2\in X^*$, так что $\varphi(x_1)=\varphi(x_2)=Z^*$, тогда:

$$Z^*\leq\varphi(x(\alpha)),\leq(1-\alpha)\varphi(x_1)+\alpha\varphi(x_2)= Z^* \qquad (2.6.11)$$

Значит, $x(\alpha)\in X^*$ и, следовательно, X^* -выпукло. Если функция $\varphi(x)$ – непрерывна, то множество X^* будет и замкнутым. Если выполняется условие *(2.6.9)*, то из *(2.6.3)* сразу же следует, что $x^*\in X^*$ и, наоборот.

Пусть $x^*\in X^*$ и допустим, что

$$\lim_{\alpha\to 0}\{[\varphi(x^*+\alpha(x-x^*)) - \varphi(x')]/\alpha\}= \varphi'(x^*)(x-x^*)<0, \qquad (2.6.12)$$

тогда, в силу непрерывности предельного перехода, будем иметь:

$$\varphi\,(x^*+\alpha(\,x-x^*))<\varphi\,(x^*)\,, \qquad (2.6.13)$$

при всех достаточно малых α, а это противоречит условию $x^*\in X^*$. Теорема доказана.

Следствие 2.6.1. *Если $x^*\in[a,b]$ – внутренняя точка минимума функции $\varphi(x)$ на отрезке $[a,b]$, то $\varphi'(x^*)=0$.*

В этом случае разность $x - x^*$ может быть и положительной и отрицательной, поэтому для выполнения условия (2.6.9) необходимо, чтобы $\varphi'(x^*)=0$.

Следствие 2.6.2. *Если $\varphi'(a_k)<0$, $\varphi'(b_k)=0$, то промежуток $[a_k, b_k]$ локализует множество оптимальности X^* , т.е. $X^*\subset[a_k, b_k]$.*

Действительно, по критерию оптимальности первого порядка выпуклая функция является унимодальной. Условие $\varphi'(a_k)<0$ означает, что функция $\varphi(x)$ в достаточно малой окрестности $O(a_k)$ убывает, поэтому $a_k<a^*$, аналогично $b_k>b^*$, так что

$$X^* = [a_k^*, b_k^*] \subset [a_k, b_k].$$

2.6.3. Метод Больцано (или метод деления отрезка пополам)

Пусть $\varphi(x)$ – выпуклая дифференцируемая функция, заданная на промежутке $[a, b]$. В рассматриваемом методе очередной промежуток $[a_{k-1}, b_{k-1}]$ делится пополам средней точкой $x_k=\frac{1}{2}(a_{k-1}+b_{k-1})$, вычисляется производная $\varphi'(a_k)$ и по её знаку выбирается новый промежуток $[a_k, b_k]$, локализующий x^* (рис. 2.6.3).

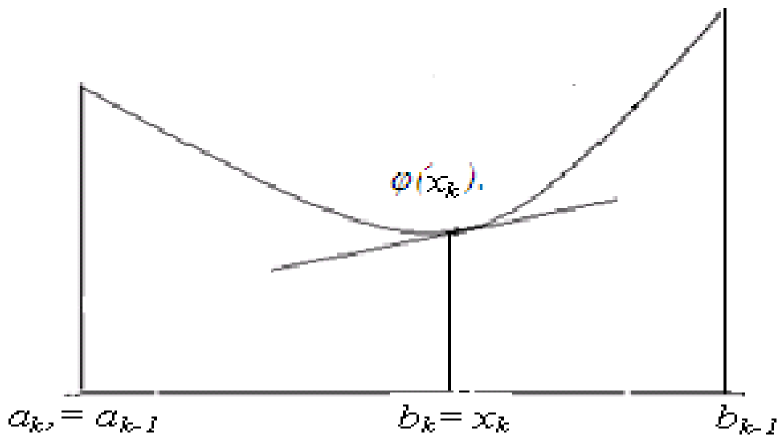

Рис. 2.6.3. Разбиение промежутка $[a_{k-1}, b_{k-1}]$ в методе Больцано.

Перед началом выполнения алгоритма следует задать начальные значения a ,b, ε. Затем следует алгоритм метода Больцано (или метода деления отрезка пополам).

Алгоритм метода деления отрезка пополам.

Задаем значения a ,b, ε.

Шаг 1. Полагаем $a_0=a$ и вычисляем $\varphi'(a_0)$. Проверяем:

- если $\varphi'(a_0)\geq 0$, то задача решена и $x^*= a_0$, $\varphi^*=\varphi(x^*)$.
- если $\varphi'(a_0)\leq 0$, полагаем $b_0=b$ *и* вычисляем $\varphi'(b_0)$.
- если $\varphi'(b_0)\leq 0$, то задача решена *и* $x^*= b_0$, $\varphi^* = \varphi(x^*)$.

*П*олагаем $k=1$ и переходим к **Шагу 2.**

Шаг 2. Вычисляем x_k по формуле:

$$x_k = 1/2 (a_{k-1} + b_{k-1}) \qquad (2.6.14)$$

Находим $\varphi'(x_k)$. Если $\varphi'(x_k)=0$, то задача решена и $x^*= x_k$, $\varphi^* = \varphi(x^*)$.

Иначе:

a) если $\varphi'(x_k)>0$, то $a_k = a_{k-1}$, $b_k =x_k$;

b) если $\varphi'(x_k)<0$, то $a_k = x_k$, $b_k = b_{k-1}$

Переходим к шагу 3.

Шаг 3 . Если $a_k - b_k < \varepsilon$, то процесс окончен, полагаем $x_k^* = 1/2(a_k + b_k)$, $\varphi^*=\varphi(x^*)$, иначе полагаем $k = k+1$ и переходим к **Шагу 2.**

Таким образом, алгоритм вырабатывает систему вложенных промежутков $[a_k, b_k] \subset [a_{k-1}, b_{k-1}]$, локализующих оптимальную точку x^*.

Если окончание процесса происходит по условию шага 3, то промежуток $[a_k, b_k]$ содержит всё множество оптимальности функции $\varphi(x)$: $X^* \subset [a_k, b_k]$.

Длина k-го промежутка $b_k - a_k = 1/2^k(b_0 - a_0)$. Поэтому погрешность приближенного решения оценивается по формуле:

$$/ x_k^* - x^*/ \leq 1/2^{k-1}(b_0 - a_0) \tag{2.6.15}$$

Теорема о сходимости методов первого порядка представлена в пункте 2.6.5.

2.6.4. Метод касательных

Предположим, что $\varphi(x)$ - выпуклая дифференцируемая функция в промежутке $[a,b]$. В методе касательных. Разбиение промежутка производится точкой x_k, в которой пересекаются касательные (2.6.16) к графику функции $\varphi(x)$ в концевых точках отрезка $[a_k, b_k]$.

$$\begin{cases} \varphi(a_k) + \varphi'(a_k)(x - a_k) = 0 \\ \varphi(b_k) + \varphi'(b_k)(x - b_k) = 0 \end{cases} \tag{2.6.16}$$

Решая систему уравнений (2.6.16) относительно x, получим:

$$x = \{[\varphi'(b_k)b_k - \varphi(b_k)] - [\varphi'(a_k)a_k - \varphi(a_k)]\}/[\varphi'(b_k) - \varphi'(a_k)] \tag{2.6.17}$$

или

$$x_{k+1} = \{[\varphi'(b_k)b_k - \varphi(b_k)] - [\varphi'(a_k)a_k - \varphi(a_k)]\}/[\varphi'(b_k) - \varphi'(a_k)], \tag{2.6.18}$$

или

$$x_k = \{[\varphi'(b_{k-1})b_{k-1} - \varphi(b_{k-1})] - [\varphi'(a_{k-1})a_{k-1} - \varphi(a_{k-1})]\}/[\varphi'(b_{k-1}) - \varphi'(a_{k-1})] \tag{2.6.19}$$

Соответствующее значение функции в точке пересечения x_k может быть вычислено по одной из формул

$$u_k = \varphi(x_к) = \varphi(a_{k-1}) + \varphi'(a_{k-1})(x_k - a_{k-1}) = \varphi(x_k) = \varphi(b_{k-1}) + \varphi'(b_{k-1})(x_k - b_{k-1}) \tag{2.6.20}$$

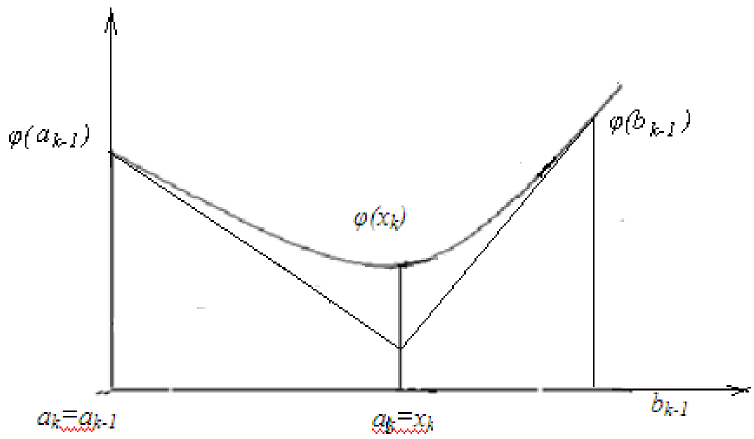

Рис. 2.6.4. Разбиение промежутка $[a_{k-1}, b_{k-1}]$ в методе касательных.

Алгоритм метода касательных формулируется также как метод Больцано, только x_k на **Шаге 2** вычисляется теперь по формуле (2.6.19) или (2.6.18), соответственно $u_k=\varphi(x_к)$ вычисляется по формуле (2.6.20). В качестве множества оптимальности аргумента x^* выбирается множество $X^*=[a_k^*, b_k^*]$, при этом $\varphi^*(x_к^*)$ – минимальное значение функции $\varphi(x)$ на отрезке $[a, b]$.

2.6.5.С ходимость методов первого порядка

Рассмотрим условия сходимости алгоритмов, реализующих первого порядка. Сходимость методов первого порядка доказывает следующая теорема.

Теорема 2.6.3. *Алгоритм, реализующий методы первого порядка, вырабатывает пробы* $\{x_к, \varphi(x_к), \varphi'(x_к)\}$, *оценки* $\{u_k=\varphi(x_к)\}$ *и вложенные промежутки* $\{[a_k, b_k]\}$ *такие, что*

$$u_k \le \varphi^*(x_к^*) \le \varphi(x_к);\ [a^*, b^*] \subset [a_k, b_k],\ k=0,1,2,\dots . \qquad (2.6.21)$$

В случае бесконечного процесса будем иметь:

$$a_k \nearrow a^*,\ b_k \searrow b^*;\quad \varphi(a_к) \searrow \varphi^*(x_к^*),\ \ \varphi(b_к) \searrow \varphi^*(x_к^*); \qquad (2.6.22)$$

$$\varphi'(a_к) \nearrow 0;\ \varphi'(a_k) \searrow 0;\ u_k \nearrow \varphi^*(x_к^*);\ \varphi(x_к) \to \varphi^*(x_к^*). \qquad (2.6.23)$$

<u>Доказательство</u>

Используя формулу (2.6.19) можно показать, что $a_k < x_{k+1} < b_k$, $k=0,1,2,\ldots$. Откуда следует, что согласно шагу 2 алгоритма, последовательность $\{a_к\}$ не убывает, а последовательность $\{b_k\}$ - не возрастает.

Поэтому $[a_k, b_k] \subset [a_{k-1}, b_{k-1}]$, $k \geq 1$.

Множество $X^*=[a_k^*, b_k^*] \subset [a_k, b_k]$, в силу следствия 2, так как $\varphi'(a_к) < 0$; $\varphi'(b_к) > 0$. Левое из неравенств (2.6.21) вытекает из следствия 2 и определения u_k по формулам (2.6.20), а правое неравенство является очевидным.

В случае бесконечного процесса ограниченные, монотонные последовательности $\{a_к\}$ и $\{b_k\}$ сходятся, причем: $a_k \nearrow a'$, $b_k \searrow b'$. Соответствующие последовательности $\{\varphi(a_к), \varphi(b_к)\}$ являются невозрастающими и ограниченны снизу значением $\varphi^*(x_к^*)$, поэтому они сходятся. В силу непрерывности $\varphi(x)$ будем иметь

$$\varphi(a_к) \searrow \varphi(a'), \quad \varphi(b_к) \searrow \varphi(b').$$

Покажем, что последовательность оценок $\{ u_k \}$строго возрастает. Действительно, при $a_k= a_{k-1}$, $b_k= x_k$, используя формулу (2.6.20) , находим (см. рис.2.6.5).

$$u_{k+1} - u_k= \varphi'(a_к)\,(x_{k+1} - x_k) > 0, \tag{2.6.24}$$

так как $x_{k+1} > x_k$, и $\varphi'(b_к) > 0$.

Откуда следует, что $u_{k+1} > u_k$ при всех k , а также $u_k \leq \varphi^*(x_к^*)$ в силу (2.6.21). При этом из монотонности и ограниченности последовательности $\{u_k\}$ следует, что она сходится к некоторому пределу $u^* \leq \varphi^*(x_к^*)$

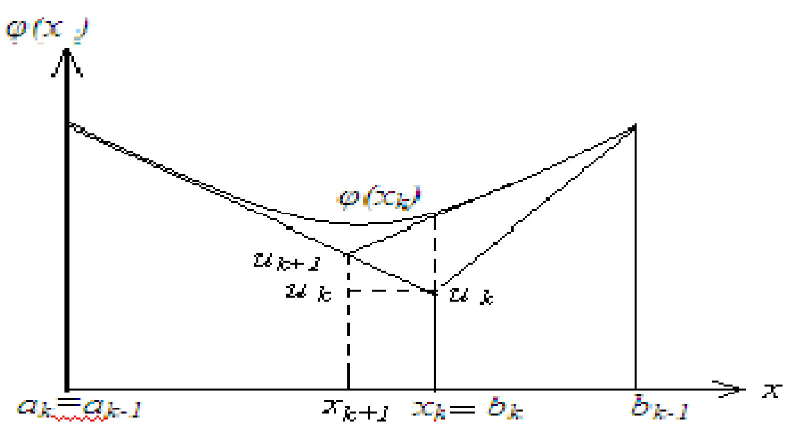

Рис. 2.6.5. Оценки u_k и u_{k+1} , в случае, $a_k = a_{k-1}$. $b_k = x_k$,.

Если $x_{k+1} = a_{k-1}$, то формулы (2.6.20) можно представить в виде

$$u_{k+1} = \varphi'(a_k)(a_{k-1} - a_k) + \varphi(a_k). \qquad (2.6.25)$$

Переходя к пределу, получим $u^* = \varphi(a')$. Аналогично, при $x_{k+1} = b_{k-1}$, будем иметь $u^* = \varphi(b')$. Следовательно,

$$u^* = \varphi(a') = \varphi(b'). \qquad (2.6.26)$$

Концы интервала a_k , b_k , обновляются бесконечное число раз. Допустим противное: $a_k = a_1$ для всех $k>s$, тогда $b_k = x_k$ при $k \geq s$ и, переходя к пределу в (2.6.20) , получим:

$$\varphi'(a_s)(b - a_s) + \varphi(a_s) = \varphi(b)$$

Для выпуклой функции это возможно лишь в том случае, когда она линейна на промежутке $[a_s, b]$. Но тогда $\varphi'(b - 0) = \varphi'(a_s) < 0$, а $\varphi'(b + 0) \geq 0$, что противоречит условию гладкости функции $\varphi(x)$. Воспользуемся теперь условием (2.6.21) и свойством выпуклости

$$u_k < z^* < \varphi(x_k) \leq (1-\alpha_k)\varphi(a_k) + \alpha_k \varphi(b_k), \quad \text{где} \quad \alpha_k = (x_{k+1} - a_k)/(b_k - a_k)$$

Переходя к пределу с учетом (2.6.22), находим $u_k = z^* = \varphi(a) = \varphi(b) = \lim \varphi(x_k)$. Откуда следует, что $a=a^*, b=b^*$. Тогда $\varphi'(a^*) = 0$, $\varphi'(b^*) = 0$ при этом $\varphi'(a_k) \nearrow 0$, $\varphi'(b_k) \searrow 0$ в силу монотонности производной выпуклой функции. Теорема полностью доказана.

2.7. Методы второго порядка

К классу методов второго порядка относятся методы, в которых явно или неявно используются первые и вторые производные целевой функции. Для вычисления второй производной требуется, как минимум, три узловых значения целевой функции $\varphi(x)$.

Поэтому к методам второго порядка в более широком смысле можно отнести все методы поиска, в которых для определения k-го приближения используется три узловых значения целевой функции.

Преимущество этих методов заключается в быстрой сходимости. Однако для их применения необходимо иметь достаточно малый промежуток локализации точки экстремума (точность решения и быстрота сходимости метода зависит от выбора начального приближения). Поэтому методы второго порядка редко используются самостоятельно, а чаще на последнем этапе для уточнения решения. Кроме того, алгоритмы этих методов достаточно сложны, что затрудняет их реализацию.

2.7.1. Критерии оптимальности второго порядка

Необходимые и достаточные условия существования точки минимума унимодальной дважды дифференцируемой функции $\varphi(x),\ x\in[a,\ b]$ определяются следующей теоремой.

Теорема 2.7.1. (*О необходимом и достаточном условии существования минимума дважды дифференцируемой функции $\varphi(x)$ для $x\in[a,\ b]$, прямая и обратная теоремы).*

Если x^*- точка локального минимума функции $\varphi(x)$ на отрезке $[a,\ b]$, то в этой точке выполняются следующие условия:

$$\mathbf{1.}\ \varphi'(x^*)(x-x^*)\geq 0,\ \forall x\in[a,b], \tag{2.7.1}$$

$$\mathbf{2.}\ \varphi''(x^*)\geq 0, \quad \text{если} \quad \varphi'(x^*)=0. \tag{2.7.2}$$

Обратно:

Если в некоторой точке $x^*\in[a,b]$ выполняется условие (2.6.27) и $\varphi''(x^*)\geq 0$ для $x\in O(x^*)$, при $\varphi'(x^*)=0$, то x^*- точка локального минимума функции $\varphi(x)$ на отрезке $[a,b]$.

<u>Доказательство</u>.

Разложим функцию $\varphi(x)$, *в точке x^* в ряд Тейлора, получим:*

$$\varphi(x)-\varphi(x^*)=\varphi'(x^*)(x-x^*)+1/2[\varphi''(x^*)(x-x^*)^2]+O(|x-x^*|) \quad (2.7.3)$$

Из (2.7.3) следует первое утверждение теоремы, так как знак в правой части выражения (2.7.3) определяется знаком первого отличного от нуля члена.

Если $\varphi'(x^*)(x-x^*)>0$, то $\varphi(x)-\varphi(x^*)>0$ в некоторой окрестности $O(x^*)$, то x^* - точка локального минимума.

Если $\varphi''(x^*)\geq 0$ *для* $x\in O(x^*)$, то из условия *(2.6.5)* теоремы 2.6.1. следует, что $\varphi(x)$ выпукла в окрестности $O(x^*)$. Но тогда условие $\varphi'(x^*)=0$ является достаточным в силу теоремы 2.6.2 для того, чтобы x^* была точкой локального минимума функции $\varphi(x)$ на отрезке $[a, b]$.

Теорема доказана.

Следствие 2.7.1. Если в некоторой точке $x^*\in[a, b]$ выполняются условия

$$\varphi'(x^*)=0 \quad \varphi''(x^*)\geq 0, \quad (2.7.4)$$

то x^*- точка изолированного локального минимума функции $\varphi(x)$.

В силу непрерывности $\varphi''(x)$ из условия (2.7.4) будем иметь $\varphi''(x^*)\geq 0$ *для* $x\in O(x^*)$. Но тогда $\varphi(x)$ - строго выпуклая функция в $O(x^*)$ и по теореме 2.7.1. она имеет в окрестности $O(x^*)$ единственную точку минимума x^*.

Условия (2.7.1) и (2.7.2) не являются необходимыми для локального минимума. Чтобы это показать рассмотрим следующий пример.

Пример 2.7.1. Рассмотрим функцию $y= x^6(1 + \sin(1/x))$, которая дважды непрерывно дифференцируема и достигает минимума в точке $x^*=0$. В этой точке $\varphi'(x^*)=0$ $\varphi''(x^*)=0$, так что выполняются условия (2.7.1), (2.7.2).

Вместе с тем, в любой окрестности вторая производная $\varphi''(x)$ меняет знаки и точка x^* является точкой сгущения локальных минимумов $x_k^*=1/((3/2+2k)\pi)$.

Аналогично и условия (2.7.4) не являются необходимыми для изолированного локального минимума.

Функция $y= x^6(1 + \sin(1/x))$ имеет одну единственную точку минимума $x^*=0$, но не удовлетворяет условиям (2.7.1), (2.7.2) и (2.7.4).

2.7.2. Метод Ньютона.

Пусть найден некоторый промежуток локализации $[a_k,b_k]$ изолированной точки минимума x^* функции $\varphi(x)$ на отрезке $[a,b]$ одним из методов нулевого или первого порядка.

Предположим, что на отрезке $[a_k,b_k]$ функция $\varphi(x)$ является строго выпуклой и удовлетворяет условию

$$0 < m \le \varphi''(x^*) \le M < \infty \qquad (2.7.5)$$

Тогда в любой пробной точке $x_k \in [a_k,b_k]$ эта функция достаточно хорошо аппроксимируется параболой, которая получается в результате разложения функции $\varphi(x)$ в ряд Тейлора:

$$T_2(x)= \varphi(x_k) + \varphi'(x_k)(x - x_k)+(1/2)\varphi''(x_k)(x - x_k)^2$$

Следует ожидать, что и точки экстремума функции $\varphi(x)$ и параболы $T_2(x)$ будут близкими. Возьмем производную $T'_2(x)$ и приравняем её к нулю, получим:

$$T'_2(x)= \varphi'(x_k) + \varphi''(x_k)(x - x_k)=0.$$

Отсюда находим точку минимума параболы $T_2(x)$:

$$x_{k+1}=x_k - \varphi'(x_k)/ \varphi''(x_k) \qquad (2.7.6)$$

и принимаем её в качестве нового приближения к искомому экстремуму x^* функции $\varphi(x)$.

В этом и состоит метод Ньютона, который иногда называют методом квадратичной аппроксимации.

Исследование сходимости метода Ньютона рассмотрено в общем многомерном случае в работе Васильева Ф.П. «Численные методы решения экстремальных задач». М. ,1978.[3]

При реализации на ЭВМ для вычисления производных обычно применяют формулы численного дифференцирования:

$$\varphi'(x_k) = (\varphi(b_k)- \varphi(a_k))/(b_k-a_k); \qquad (2.7.7)$$

$$\varphi''(x_k)=[(b_k-x_k)\, \varphi(a_k)-(b_k-a_k)\varphi(x_k)+ (x_k-a_k)\varphi(b_k)]/[(x_k-a_k)(b_k-x_k)(b_k-a_k)]$$

Тогда, подставляя выражения *(2.7.7)* в формулу *(2.7.6),* получим рекуррентную формулу (2.7.8) для вычисления очередного значения x_{k+1}:

$$x_{k+1}=x_k-(x_k-a_k)(b_k-x_k)[\varphi(b_k)-\varphi(a_k)]/[(b_k-x_k)\varphi(a_k)-(b_k-a_k)\varphi(x_k)+(x_k-a_k)\varphi(b_k)] \quad (2.7.8)$$

В случае равноотстоящих узлов $(x_k-a_k)=(b_k-x_k)=\Delta x_k$ формула принимает вид:

$$x_{k+1}=x_k-\{\Delta x_k[\varphi(b_k)-\varphi(a_k)]\}/[\varphi(a_k)-2\varphi(x_k)+\varphi(b_k)] \quad (2.7.9)$$

Для сходимости приближений требуется выполнение условий, аналогичных условиям *(2.7.5):*

$$\varphi(a_k) > \varphi(x_k) \le \varphi(b_k) \text{ или } \varphi(a_k) \ge \varphi(x_k) < \varphi(b_k), \quad (2.7.10)$$

то есть хотя бы одно из неравенств выполняется строго.

Итак, пусть найден промежуток $[a,b]$ локализации точки минимума функции $\varphi(x)$ с помощью алгоритма, рассмотренного в п. 2.5.8.

Будем считать, что на промежутке $[a,b]$ выполняется условие (2.7.5) и выбрана пробная точка y. В точке y разложим $\varphi(x)$ в ряд Тэйлора и в качестве квадратичной аппроксимации возьмем три члена этого ряда

$$\varphi(x)=\varphi(y)+\varphi'(y)*(x-y)+1/2\varphi''(y)(x-y)^2$$

Согласно (2.7.6) минимум этого полинома достигается в точке $x=y-\varphi'(y)/\varphi''(y)$. Принимаем эту точку ($x=y-\varphi'(y)/\varphi''(y)$.) в качестве нового приближения к экстремуму x* функции $\varphi(x)$.

Для получения более точного решения положим $y=x$ и определим новое приближение. В этом и состоит метод Ньютона (метод квадратичной аппроксимации).

Если отрезок $[a,b]$ локализует точку максимума, то значение $x = y-\varphi'(y)/\varphi''(y)$ будет приближенным значением точки максимума функции $\varphi(x)$.

Этот метод быстро сходится, однако существуют случаи, когда метод расходится. Признаком расходимости является выполнение условия $x\notin[a,b]$.

Алгоритм метода Ньютона

Выберем пробную точку y на промежутке$[a,b]$($y \in [a,b]$).

Шаг1. Вычислим $Fy=\varphi(y)$; $x=y-\varphi'(y)/\varphi''(y)$;

Шаг 2. Если $x \notin [a,b]$, то идти на **Шаг 6**, иначе $Fx=\varphi(x)$.

Шаг 3. Если $|Fx - Fy| > \varepsilon$ то идти на **Шаг 4,** иначе переход на **Шаг 5.**

Шаг 4. Положим $y=x$, $Fy= Fx$ и перейдем на **Шаг1.**

Шаг 5. Если $Fy < Fx$, то $x^*=y$, $Fx^*= Fy$, иначе $x^*=x$, $Fx^*= Fx$. Вывод x^*, Fx^*, перейти на **Шаг 7.**

Шаг 6.. Вывод сообщения «Метод расходится».

Шаг 7. Конец.

2.7.3.Метод ДСК.

На основе метода Ньютона разработан метод ДСК, который получил свое название по инициалам его авторов – Девиса, Свена и Кемпи [1, 11].

Отличие этого метода заключается в том, что он предусматривает: выбор направления поиска экстремума, поиск промежутка локализации экстремума, регулировку шага поиска по направлению, использование формулы (2.7.8) для определения очередного приближения и условия сходимости (2.7.10).

Рассмотрим упрощенный алгоритм ДСК, основанный на формулах (2.7.8) для неравноотстоящих узлов.

Алгоритм метода ДСК.

Задаем ε, x_0, Δx_0 и вычисляем $\varphi(x_0)$. Полагаем $k=0$.

Шаг 1. Находим, $x_{k+1}=, x_k+\Delta x_k$ и вычисляем $\varphi(x_{k+1})$.

Шаг 2. Если $\varphi(x_{k+1}) \le \varphi(x_k)$, то полагаем $\Delta x_{k+1} = 2\Delta x_k$ и переходим к **Шагу 1**, иначе переходим к **Шагу 3**.

Шаг 3. Если $\varphi(x_{k+1}) > \varphi(x_k)$ и $k=0$, то полагаем $x_0,= x_1$, $x_1= x_0$, $\Delta x_1=- 2\Delta x_0$ и переходим к **Шагу 1,** иначе переходим к **Шагу 4**.

Шаг 4. Если $\varphi(x_{k+1})> \varphi(x_k)$ и $k>0$, то полагаем $a_k=x_{k-1,}$ $b_k=x_{k+1}$, *если* $\Delta x_k > 0$, $a_k=x_{k+1}$, $b_k=x_{k-1}$, *если* $\Delta x_k < 0/ x_0, x_0$,

Шаг 5. Вычисляем x_{k+1} по формуле (2.7.8), затем значение $\varphi(x_{k+1})$.

Шаг 6. Если $/\varphi(x_{k+1})$- $\varphi(x_k)/< \varepsilon$, то процесс окончен и полагаем $x^*= x_{k+1}$, $z^*=/\varphi(x_{k+1})$, иначе переход на **Шаг 7**.

Шаг 7. Если $x_{k+1} < x_k$, то полагаем $x_k= x_{k+1}$, $\varphi(x_k)= \varphi(x_{k+1})$, $x_{k+1}= x_k$, , $\varphi(x_{k+1})= \varphi(x_k)$.

Шаг 8. Определяем новый промежуток локализации x^*:

$a_{k+1}=a_k$; $x_{k+1}=x_k$;, $b_{k+1}=x_{k+1}$, если $\varphi(x_k)\leq \varphi(x_{k+1})$,

$a_{k+1}=x_k$; $x_{k+1}=x_{k+1}$;, $b_{k+1}=b_k$, если $\varphi(x_k)> \varphi(x_{k+1})$, и переходим к **Шагу 5**.

***Пояснения к алгоритму*:**

- **шаги 1 – 4** осуществляют поиск промежутка локализации x^*. В случае удачной пробы **на шаге 2** величина Δx_k удваивается и ведется дальнейший поиск.

- при неудачной пробе и $k=0$ производится изменение направления поиска (**шаг 3**). При неудачной пробе и $k>0$ производится поиск промежутка локализации x^*. На рис.2.7.1.. это будет промежуток $[a_3, b_3]= [x_2, x_4]$.

- **шаги 5 – 8** реализуют непосредственно метод Ньютона с использованием формулы (2.7.8) и условия (2.7.10).

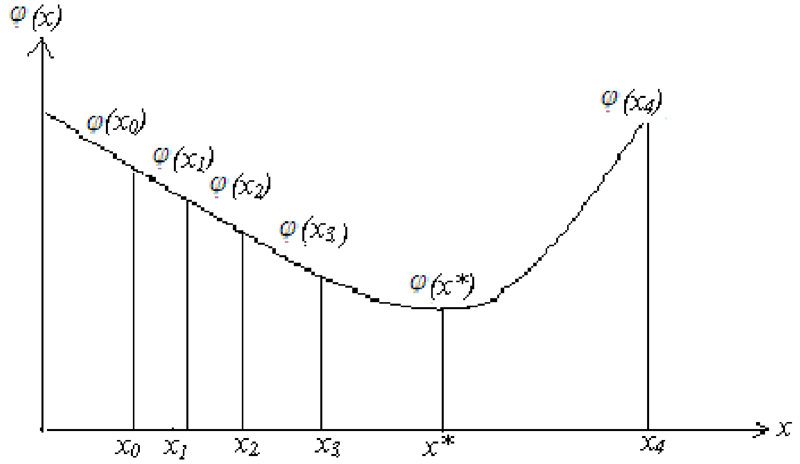

Рис.2.7.1. Поиск промежутка локализации x^* методом ДСК.

2.7.4. Метод квадратичной интерполяции.

Вместо квадратичной аппроксимации функции $\varphi(x)$ рядом Тейлора можно использовать квадратичную интерполяцию.

Пусть имеются три узловые точки x_0, x_1, x_2,удовлетворяющие соотношению $x_0 < x_1 < x_2$. Для них вычислены соответствующие значения φ_0, φ_1, φ_2 целевой функции $\varphi(x)$. Построим интерполяционный многочлен Лагранжа:

$$L(x)=\{[(x-x_1)(x-x_2)]/[(x_0-x_1)/(x_0-x_2)]\}\varphi_0+\{[(x-x_0)(x-x_2)]/[(x_1-x_0)/(x_1-x_2)]\}\varphi_1+ +\{[(x-x_0)(x-x_2)]/[(x_2-x0)/(x2-x1)]\}\varphi_2. \quad (2.7.11)$$

Найдем его производную:

$$L'(x)=\{[2x-(x_1+x_2)]/[(x_0-x_1)/(x_0-x_2)]\}\varphi_0+\{[2x-(x_0+x_2)]/[(x_1-x_0)/(x_1-x_2)]\}\varphi_1+ +\{[2x-(x_0+x_1)]/[(x_2-x_0)/(x_2-x_1)]\}\varphi_2.$$

Из условия $L'(x)=0$ определяем приближенную точку экстремума:

$$\tilde{x}=-1/2\{[(x_2^2-x_1^2)\varphi_0-(x_2^2-x_0^2)\varphi_1+(x_1^2-x_0^2)\varphi_2]\}/\{[(x_2-x_1)\varphi_0-(x_2-x_0)\varphi_1+(x_1-x_0)]\varphi_2\} \quad (2.7.12)$$

Если выполняется условие (2.6.36), то:

$$\varphi_1 \le \min\{\varphi_0, \varphi_2\}, \quad (2.7.13)$$

и, следовательно эта точка будет принадлежать интервалу $[x_0, x_2]$.

Алгоритм метода квадратичной интерполяции.

Задаем точки $x_0 < x_1 < x_2$. и вычисляем $\varphi_0=\varphi(x_0)$, $\varphi_1=\varphi(x_1)$, $\varphi_2=\varphi(x_2)$. Проверяем условие (2.7.13). Если оно не выполняется, то точки x_0, x_1, x_2 выбраны неверно и алгоритм прекращает свою работу. Иначе переходим к **Шагу 1**.

Шаг 1. Определяем $\tilde{x}$ по формуле (2.7.12) и вычисляем $\tilde{\varphi}=\varphi(\tilde{x})$.

Шаг 2. Если $|\varphi(\tilde{x}).-\varphi(x)|<.\varepsilon$, то процесс окончен, иначе из точек x_0, x_1, x_2, $\tilde{x}$ выбираем тройку, удовлетворяющую условию (2.7.13), присваиваем ей обозначения x_0, x_1, x_2 и переходим к **Шагу 1**.

Преимущество этого метода сравнительно с методом Ньютона состоит в том, что для его применения не требуется строгая выпуклость целевой функции, а достаточно лишь условие унимодальности. Однако условие (2.7.10) должно соблюдаться на каждой итерации.

Если в алгоритме метода ДСК заменить формулу (2.7.8) на формулу (2.7.12), то получим комбинированный алгоритм ДСК – Пауэла, который зарекомендовал себя как один из наиболее эффективных алгоритмов поиска экстремума.[1, 11]

2.8. Многоэкстремальные задачи.

При решении практических задач точно установить свойства унимодальности и выпуклости функций часто не представляется возможным, так как не существует эффективных критериев унимодальности и выпуклости функций. Единственность экстремума в отдельных случаях может быть установлена исходя из физической сущности моделируемого процесса. Многие задачи оптимального планирования и проектирования по своей природе являются многоэкстремальными. Если применять для решения таких задач один из рассмотренных методов, то будет найден какой-либо локальный экстремум, который может значительно отличаться от глобального. Поэтому для решения многоэкстремальных задач разработаны специальные методы.

Полная классификация и характеристика этих методов представлена в работах [3] и [6]. Все эти методы отличаются высокой сложностью и требуют большого объема оперативной памяти при проведении вычислений на ЭВМ, что значительно снижает их быстродействие. В работе рассмотрен наиболее компактный метод глобального поиска, который построен Тетеревым А. Г. [1] на основе синтеза метода ветвей и границ и метода ломаных [3].

2.8.1. Локализация экстремумов. Метол сканирования.

В общем случае функция $\varphi(x)$ на множестве E имеет несколько точечных и /или интервальных экстремумов (рис.2.8.1). Среди них содержатся и глобальные экстремумы. Согласно классификации экстремумов (таблицы 2.1.1, 2.1.2) существует несколько классов экстремальных задач. Рассмотрим наиболее общую из них: задачу определения всех минимумов функции $\varphi(x)$ на отрезке $[a,b]$.

Будем предполагать, что этот промежуток ограничен. Наиболее простой подход к решению задачи состоит в локализации точек и интервалов оптимальности с последующим применением методов поиска локальных экстремумов.

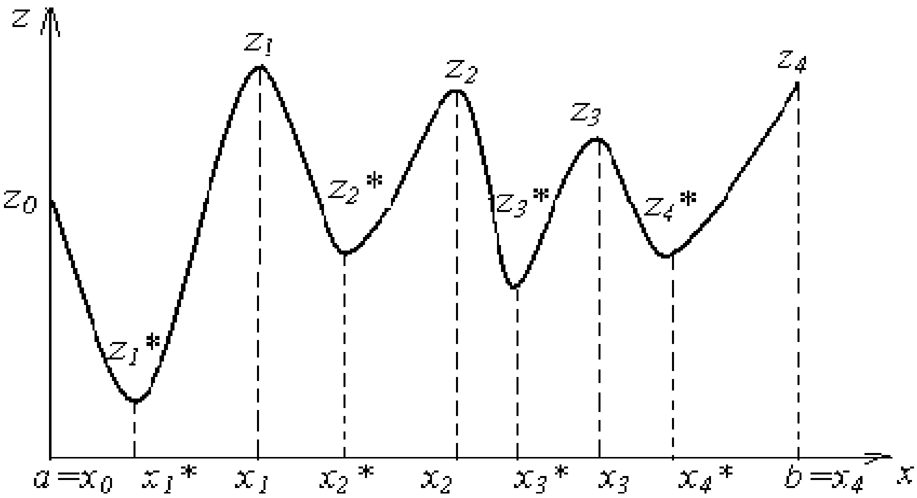

Рис.2.8.1. Локализация экстремумов.

Разобьем множество $E=[a,b]$ на n промежутков $[x_{k-1}, x_k]$, $k=1,2,3,\dots,n$ узловыми точками $x_k=a+k\Delta x_0$, где $k=0,1,2,\dots,n$, $x_0=a$, $x_n=b$. Вычислим все значения $\varphi(x_k)$, $k=0,1,2,\dots,n$. Затем рассмотрим скользящую по отрезку $[a,b]$ тройку узлов x_{k-1}, x_k, x_{k+1}. Если выполняются неравенства

$$z_{k+1} > z_k < z_{k+1}, \tag{2.8.1}$$

то интервал (x_{k-1}, x_{k+1}) содержит хотя – бы одну точку минимума. В этом легко убедиться, анализируя возможные варианты хода кривой $z = \varphi(x)$ на отрезке $[x_{k-1}, x_{k+1}]$. В граничных узлах $x_0=a$, $x_k=b$ должно выполняться одно из неравенств (2.8.2):

$$z_0<z_1,\ \ z_n<z_{n+1} \tag{2.8.2}$$

В каждом интервале локализации точки минимума дальнейший поиск можно вести методом локальной оптимизации. В результате будет получено частичное решение задачи.

Во-первых, при заданном Δx близко лежащие друг от друга экстремумы могут оказаться неразделенными. Поэтому в каждом промежутке мы найдем лишь одну точку экстремума, а все остальные будут потеряны. Например, на рис. 2.8.1. выполняется условие $z_2 > z_3 < z_4$, значит промежуток $[x_2, x_4]$ содержит точку минимума. Но, в действительности, он содержит два минимума (x_3^*, z_3^*) и (x_4^*, z_4^*).

Во-вторых, некоторые экстремумы будут также потеряны, если они проявляются лишь в пределах одного промежутка $[x_{k-1}, x_k]$.

В методе сканирования описанная процедура локализации повторяется систематически. На каждой итерации шаг Δx уменьшается в несколько раз, то есть $\Delta x = \alpha\Delta x$ $(\alpha<1)$, следовательно производится измельчение всех промежутков $[x_{k-1}, x_{k+1}]$ локализации экстремума, вычисляются значения функции $\varphi(x_k)$ в новых узлах и производится дифференциация и уточнение интервалов локализации экстремума. В пределе интервалы стягиваются к точкам $x^* \in X^*$. В случае задачи максимизации, условие (2.8.1) заменяется условием (2.8.3):

$$z_{k+1} < z_k > z_{k+1} \qquad (2.8.3)$$

2.8.2. Поиск глобального экстремума. Метод глобального поиска.

Рассмотрим задачу определения всех глобальных минимумов функции $\varphi(x)$ на ограниченном промежутке $[a, b]$. Для её решения применим метод глобального поиска [3]. В основе этого метода лежит хорошо известный метод ветвей и границ.

Полагаем $x_0=a$, $x_1=b$, $z_0= \varphi(x_0)$, $z_1= \varphi(x_1)$ и отрезок $\Delta_1 = [x_0, x_1]$ принимаем за начальную вершину дерева поиска при $k=1$(рис.2.8.2).

Определяем пробную точку ξ_1, вычисляем локальную оценку u_1, глобальные оценки $v_1= u_1$ и $w_1= min\{z_0,z_1\}$ и производим разбиение Δ_1 на два интервала Δ_1 и Δ_2.

На k-ом шаге будут получены узлы $x_0, x_1, \ldots, x_k$, отрезки $\Delta_i = [x_i, x_j]$, $i=1,2,,\ldots,k$, значения целевой функции z_i, z_j , пробные точки $\xi_i \in \Delta_i$, локальные оценки u_1 ,

$$u_1 < \varphi(x)\ ,\ x \in \Delta_i \tag{2.8.4}$$

и глобальные оценки

$$v_k = min\{\ u_i / \ i=1,2,,\ldots,k,\ \}\ ,\ w_k=min\{z_i / \ i=0,1,2,,\ldots,k,\ \} \tag{2.8.5}$$

Из (2.8.4) и (2.8.5) следует, что при всех k будет выполняться неравенство

$$v_k < z^* \leq w_k \tag{2.8.6}$$

При этом последовательность { w_k }. в силу определения, является невозрастающей, а оценки u_i строятся так, что последовательность { v_k } будет неубывающей.

Разбиению подлежит интервал Δp с наименьшей оценкой $u_p = v_k$ /Неперспективные интервалы Δi,

для которых $u_i \geq w_k$ отсеиваются.

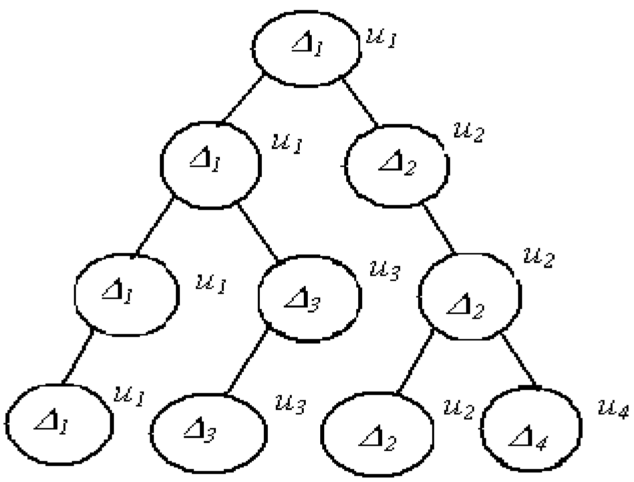

Рис. 2.8.2. Дерево поиска глобального минимума.

Процесс поиска заканчивается по заданной точности, если выполняется неравенство (2.8.7).

$$w_k - v_k < \varepsilon \tag{2.8.7}$$

В бесконечном варианте критерием оптимальности служит условие

$$lim\ (w_k - v_k\ )= 0 \tag{2.8.8}$$

На выходе получаем систему замкнутых промежутков $\{\Delta_i\}$, содержащих все точки глобального минимума и соответствующие характеристики этих промежутков. Рассмотрим основные операции метода.

Формирование оценок. Будем предполагать, что целевая функция $\varphi(x)$ удовлетворяет условию Липшица:

$$/\,\varphi(x) - \varphi(x')/ \le L\,/x - x'/,\ x \in [a, b] \tag{2.8.9}$$

В случае дифференцируемой функции это условие означает, что производная $\varphi'(x)$ ограничена

$$/\,\varphi'(x)/ \le L, \quad x \in [a, b] \tag{2.8.10}$$

На k-ом шаге процесса рассмотрим кусочно-линейную функцию (рис.2.8.3):

$$u\,(x/\alpha)= max\,[z_i - \alpha\,(x_i - x)]\ , \qquad i=0,1,2,\ldots,k \tag{2.8.11}$$

Из условия Липшица следует, что при $\alpha > L$ она минорирует целевую функцию $\varphi(x)$:

$$u\,(x/\alpha)< \varphi(x), \quad x \in \Delta_i \tag{2.8.12}$$

причем в узловых точках имеем: $u\,(x_i/\alpha)< \varphi(x_i)$. В любом промежутке $\Delta_i = [x_i, x_j]$ функция $u\,(x/\alpha)$ имеет единственную точку минимума:

$$\xi_i = (x_i + x_j)\,/2 \ + \ (z_i - z_j)\,/\,2\alpha \tag{2.8.13}$$

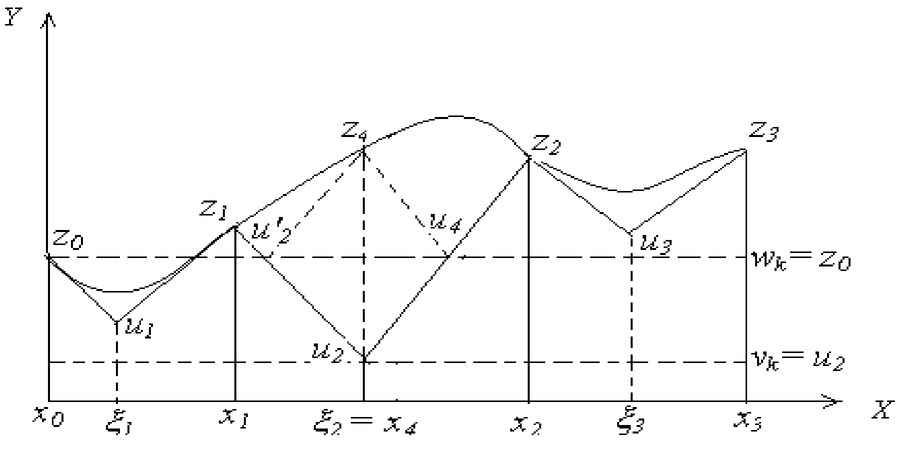

Рис.2.8.3. Формирование оценок

В этой точке она принимает значение:

$$u_i = ½[\alpha(x_i - x_j) \ + \ (z_i + z_j)], \tag{2.8.14}$$

которое в силу (2.8.12) можно принять в качестве локальной оценки

$$u_i< \varphi(x), \quad x \in \Delta i \tag{2.8.15}$$

Разбиение интервала. Рассмотрим одну из основных операций метода – операцию разбиения интервалов. На k-ом шаге разбиению подвергается отрезок $\Delta_0 =[x_0, x_q]$ с наименьшей оценкой $v_k=u_p$ (рис.2.8.4). При этом

$$x_{k+1} = \xi_p, \; z_{k+1}= \varphi(x_{k+1}).$$

Пробные точки ξ_{k+1} и ξ'_p в новых промежутках $\Delta_{k+1}=[x_{k+1}, x_q]$ и $\Delta'_p = [x_p, x_{k+1}]$

симметричны относительно x_{k+1}.Ппоэтому оценки u'_p и u_{k+1} всегда совпадают. Можно показать, что

$$\xi_{k+1} = \xi_p+ 1 / 2\alpha(z_{k+1}- v_k), \tag{2.8.16}$$

$$u_{k+1} =1/2(z_{k+1}+ v_k) , \tag{2.8.17}$$

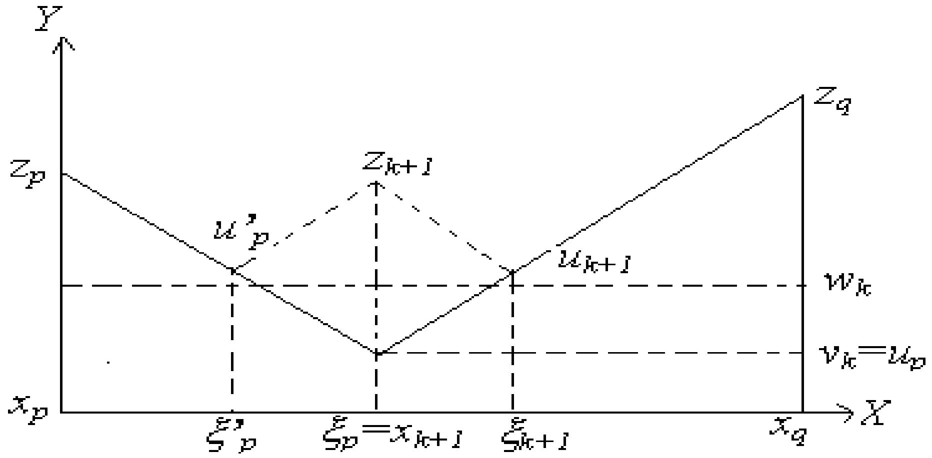

Рис.2.8.4. Схема разбиения интервалов

Если $u_{k+1} > w_k$, то новые интервалы будут неперспективными. Условие (2.8.17), равносильно следующему условию:

$$z_{k+1} >2 w_k - v_k \tag{2.8.19}$$

Это условие можно проверять, не производя самого разбиения. Если это условие выполняется, то интервал Δ_p сразу аннулируется и З переход к следующему шагу. Если же условие (2.8.19) не выполняется, то интервал Δ_p аннулируется после разбиения и на его место помещается интервал Δ'_p, которому снова присваивается обозначение Δ_p.

Из формул (2.8.16)–(2.8.19) следует, что вся необходимая информация содержится в пробных точках ξ_i, локальных оценках u_i и глобальных оценках v_k и w_k. Дополнительно, требуется лишь вычисление одного значения z_{k+1} целевой функции в наиболее перспективной точке $x_{k+1}=\xi_p$. Концевые точки x_i, x_j интервалов $\Delta_i=(x_i, x_j)$, а также значения z_i, z_j целевой функции $\varphi(x)$ в этих точках в явном виде не используются и могут не храниться. Они нужны только на начальной итерации отрезка $[x_0, x_1]$.

<u>Анализ работы метода.</u> Пусть метод прекращает работу на k-ом шаге по условию (2.8.7). На этом шаге будем иметь некоторую систему промежутков Δ_i, $i \in J_k$ таких, что $y_k \le u_i < w_k$ (рис.2.8.5). Точки глобального минимума могут размещаться лишь в интервале $\Delta_i=(x_i,x_j)$, определяемом условием $u(x/\alpha) < w_k$

$$x_j' = \xi_i - 1/\boldsymbol{\alpha}\,(w_k - u_i), \quad x_j' = \xi_i + 1/\boldsymbol{\alpha}\,(w_k - u_i)\,. \qquad (2.8.20)$$

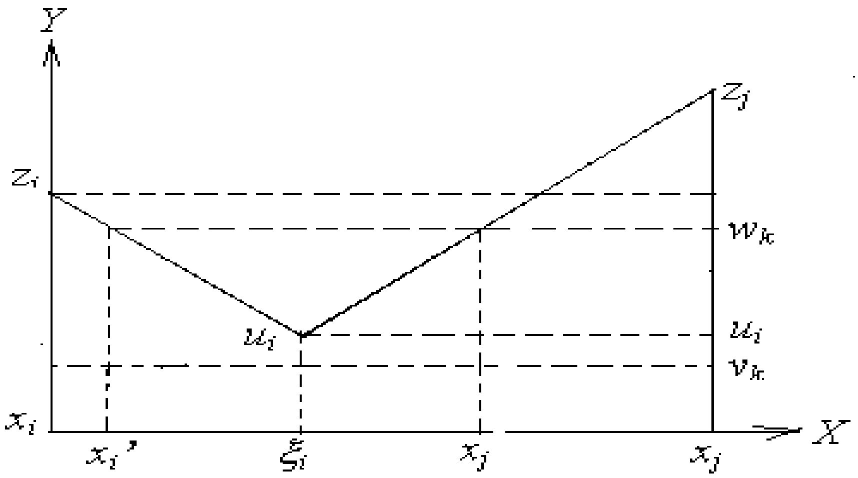

Рис.2.8.5. Анализ разбиения интервала $\Delta=[x_i,x_j]$

Следовательно, ξ_i является приближением точек глобального экстремума в интервале Δ_i, если они существуют, с погрешностью

$$\delta i = 1/\boldsymbol{\alpha}(w_k - u_i)\ , \qquad (2.8.21)$$

а w_k приближает z^* с погрешностью

$$\varepsilon_k = w_k - v_k \qquad (2.8.22)$$

2.8.3. Алгоритм метода глобального поиска

Рассмотрим формализованное описание алгоритма глобального поиска.

Шаг1. Зададим значения параметров a, b, α, ε. Присвоим $x_0=a$, $x_1=b$ и вычислим значения функции $\varphi(x)$ в этих точках: $z_0=\varphi(x_0)$, $z_i=\varphi(x_1)$. Найдем $w_1=min\{z_0,z_1\}$.Определим пробную точку ξ_1 для интервала $\Delta_1 = [x_0, x_1]$ по формуле (2.8.13) и вычислим оценку u_1 по формуле (2.8.14). Присвоим номер итерации $k:=1$ и перейдем на **Шаг 2**.

Шаг2. Найдем минимальную оценку $v_k=u_p$ по формуле (2.8.5) и присвоим значение $x_{k+1}:= \xi_p$. Вычислим $z_{k+1}=\varphi(x_{k+1})$. Если выполняется условие (2.8.19), то аннулируем пробу (ξ_p, u_p) и переходим на **Шаг 4**. Иначе проверяем условие $z_{k+1}< w_k$. Если оно выполняется, то полагаем $w_{k+1}= z_{k+1}$ и аннулируем пробы (ξ_i, u_i), для которых $u_i\geq w_{k+1}$ и переходим на **Шаг 3**. Если $z_{k+1}\geq w_k$, то полагаем $w_{k+1}= w_k$.

Шаг3. Вычисляем новую пробу (ξ_{k+1},u_{k+1}) по формулам (2.8.16)-(2.8.17) и симметричную ей пробу (ξ'_p,u'_p) по формулам (2.8.18). Аннулируем (ξ_p,u_p) и присваиваем $(\xi_p,u_p):=(\xi'_p,u'_p)$.

Шаг4. Если $w_k - v_k \leq \varepsilon_k$, то переходим к **Шагу 5**, иначе присваиваем $k:=k+1$ и переходим к **Шагу 2**.

Шаг5. Вычислим погрешность δ_i и ε_k по формулам (2.8.21) и (2.8.22) и выводим значения w_k , ε_k, ξ_i, δ_i, $i \in J_k$.

Замечание. Если условие $\alpha > L$ не выполняется, то пробная точка ξ_i может оказаться за пределами интервала Δ_i . В этом случае нужно предусмотреть остановку работы алгоритма и корректировку α. После этого возобновить работу алгоритма, начиная с шага 1. Однако, нарушение условия $\alpha > L$ может и не обнаружиться, тогда алгоритм не может гарантировать достижения глобального минимума.

2.8.4. Исследование сходимости метода глобального поиска

Для исследования сходимости метода глобального поиска докажем теорему о сходимости метода в предположении, что целевая функция удовлетворяет условию Липшица с константой L.

Теорема 2.8.1. *(о сходимости метода глобального поиска).* При $\varepsilon=0$ и $\alpha > L$ справедливы следующие утверждения:

1. Алгоритм глобального поиска вырабатывает бесконечные последовательности точек (x_k, z_k) , проб (ξ_k, u_k) и оценок (v_k, w_k).

2. В пределе выполняется условие оптимальности (2.8. 8) .

3. Любая точка сгущения x^* системы пробных точек $\{\xi_k\}$ является точкой глобального минимума $x^*\in X^*$ и наоборот, любая точка глобального минимума $x^*\in X^*$ является точкой сгущения последовательности $\{\xi_k\}$.

Доказательство.

Докажем первое утверждение: при $\alpha > L$ любая пробная точка ξ_i,является внутренней точкой соответствующего интервала Δ_i. Поэтому алгоритм глобального поиска не вырождается и вырабатывает бесконечную последовательность точек (x_k, z_k) , а значит и бесконечные последовательности (ξ_k, u_k) и (v_k, w_k).

Докажем второе утверждение:. среди интервалов $\{\Delta_k\}$ неявно порождаемых алгоритмом глобального поиска найдется бесконечная система $\{\Delta_{k1}\}$вложенных интервалов, причем каждый последующий интервал Δ_{k1+1} получается в результате разбиения предыдущего интервала Δ_{k1} , так, что локальные оценки совпадают с глобальными оценками $u_{л1}= v_{k1}$. Для них по формуле (2.8.17) будем иметь:

$$v_{k1+1} = 1/2(z_{k1}+ v_{k1}),$$

где $z_{k1}= \varphi(\xi_{k1})$.Вычитая из обеих частей этого равенства значение v_{k1}, получим

$$v_{k1+1} - v_{k1} = 1/2(z_{k1} - v_{k1}). \qquad (2.8.23)$$

Последовательность $\{v_k\}$ монотонна и ограничена, поэтому сходится. Откуда следует, что $v_{k1+1} - v_{k1} \to 0$, но тогда, в силу (2.8.23) $z_{k1} - v_{k1} \to 0$ и, тем более $w_{k1+1} - v_{k1+1} \to 0$, так как $z_{k1} \geq w_{k1+1} \geq v_{k1+1} \geq v_{k1}$. Из сходимости к нулю некоторой подпоследовательности из ограниченной и монотонной последовательности $\{w_k - v_k\}$ следует сходимость к тому же пределу и всей последовательности, то есть выполняется условие (2.8.8).

Докажем третье утверждение: в силу (2.8.6) и (2.8.8) имеем:

$$\lim w_k = \lim v_k = z^* \qquad (2.8.24)$$

Пусть x^* - точка сгущения пробных точек ξ_k , тогда найдется такая подпоследовательность$\{\xi_{ks}\}$, что $x^* = \lim \xi_{k1}$/ Без ограничения общности можно считать, что интервалы Δ_{k1} являются вложенными, поэтому для них $z_{k1} - v_{k1} \to 0$. Но тогда

$\varphi(x^*) = \lim \varphi(\xi_{k1}) = \lim z_{k1} = \lim v_{k1} + \lim(z_{k1} - v_{k1}) = z^*$ *в* силу (2.8.24). Следовательно, $x^* \in X^*$

Наоборот, пусть $x^* \in X^*$.Допустим, что x^* не является точкой сгущения последовательности $\{\xi_{ks}\}$, тогда x^* принадлежит некоторому интервалу Δ_i конечной длины. Этот интервал на всех шагах, начиная с некоторого $k=k_0$, остается перспективным, поэтому

$$\varphi(x^*) > u_i \geq v_k, \ \forall k > k_0/$$

Переходя к пределу , получим:

$$\varphi(x^*) > u_i \geq z^*,$$

что противоречит условию $x^* \in X^*$.

Следовательно, наше предположение не верно и любая точка $x^* \in X^*$ действительно является предельной для $\{\xi_k\}$.

Замечание. В третьем утверждении теоремы последовательность $\{\xi_k\}$ можно заменить последовательностью узлов$\{x_k\}$.

3. Безусловная оптимизация.

3.1. Понятие локального и глобального экстремума. Существование решения.

Различают точки минимума двух видов.

Точка $x_* \in X$ называется точкой <u>локального</u> минимума, если $f(x_*) \le f(x)$ $\forall x \in O\varepsilon(x_*)$, где $O\varepsilon(x_*) = \{x \in R_n \mid //x_* - x// \le \varepsilon\}$ - ε - окрестность точки x_*, $\varepsilon > 0$.

Точка x_* называется точкой <u>глобального</u> минимума, если $f(x_*) \le f(x)$ $\forall x \in X$.

Множество $x \in R^n$ называется компактным, если любая последовательность $\{x_k\} \in X$ имеет, хотя бы одну предельную точку $x_* \in X$. Известно, что всякая ограничённая последовательность имеет хотя бы одну предельную точку. Поэтому в R^n компактным является любое замкнутое ограниченное множество.

Следующая теорема даёт достаточные условия существование оптимального решения задачи (1)-(3).

Теорема 1 (Вейерштрасса). Для того чтобы в задаче (1)-(3) существовала точка глобального минимума, достаточно, чтобы допустимое множество X было компактно, а целевая функция f непрерывна на X.

В силу сложности проверки ограниченности множества X, а зачастую, в силу его неограниченности на практике часто применяется следствие:

Следствие (теоремы Вейерштрасса). Если функция f непрерывна в R^n и $\lim_{||x||\to\infty} f(x) = +\infty$, то f достаёт своего глобального минимума в любом замкнутом подмножестве R^n.

3.2. Необходимые сведения

3.2.1. Сведения из анализа (градиент, гессиан, локальные приближения)

1) Градиент. Линейное локальное приближение. Скалярная функция $f(x)$, $x \in R^n$ (краткая запись $f\colon R^n \to R^1$) называется дифференцируемой в точке x, если найдётся вектор $\nabla f(x)$, называемый градиентом функции такой, что

$f(x+y)=f(x)+(\nabla f(x),y)+0(y)$, $\forall\ y\in R^n$, где $0(y)$ обозначается величина, удовлетворяющая соотношению $|0(y)|/\|y\| \to 0$ при $\|y\| \to 0$.

Иначе можно сказать, что функция дифференцируема в точке x, если она допускает линейную аппроксимацию первого порядка в этой точке, т.е. найдётся линейная функция $\bar{f}(y)=f(x)+(\nabla f(x),y)$ такая, что $/f(x+y) - \bar{f}(y)/=0(y)$. Градиент определяется однозначно, при этом $\nabla f(x)=(\partial f(x)/\partial x_1,....\partial f(x)/\partial x_n)$. Величина

$$f'(x,y)=\lim_{\varepsilon\to+0}\frac{f(x+\varepsilon y)-f(x)}{\varepsilon}=(\nabla f(x),y)$$

называется производной по направлению функции $f(x)$ в точке x.

2)Вторые производные. Квадратичное представление. Скалярная функция $f(x)$ на R^n называется дважды дифференцируемой в точке x, если она дифференцируема в этой точке, и найдется симметричная n x n матрица $\nabla^2 f(x)$, называемая матрицей вторых производных (матрицей Гессе или гессиан), такая, что

$$f(x+y) = f(x) + (\nabla f(x),y) + (\nabla^2 f(x)y,y)/2 + o(\|y\|^2), \quad \forall\, y \in R^n$$

Иначе говоря, функция дважды дифференцируема в точке x, если она допускает квадратичную аппроксимацию второго порядка в окрестности этой точки, т.е. существует квадратичная функция

$$\bar{f}(y)= f(x)+(\nabla f(x),y)+(\nabla^2 f(x)y,y)/2$$

такая, что $|f(x+y)-\bar{f}(y)|=o(||y||^2)$.

3.3. Классы функций.

Функция $f(x)$, $x \in R^n$, называется **выпуклой,** если

$$f(\lambda x+(1-\lambda)y) \le \lambda f(x)+(1-\lambda)f(y),\ \forall x, y \in R^n,\ 0 \le \lambda \le 1. \quad (3.3.1)$$

Функция $f(x)$, $x \in R^n$, называется **строго выпуклой**, если в (3.3.1) при $x\neq y$ выполняется строгое неравенство. Функция $f(x)$, $x \in R^n$, называется сильно выпуклой с константой $\rho > 0$, если при $0 \le \lambda \le 1$

$$f(\lambda x + (1-\lambda) y) \le \lambda f(x) + (1-\lambda) f(y) - \rho\lambda(1-\lambda)\|x-y\|^2/2 \quad (3.3.2)$$

Для дифференцируемой функции $f(x)$, $x\in R^n$ выпуклость эквивалентна неравенству

$$f(x+y) \geq f(x) + (\nabla f(x),y), \tag{3.3.3}$$

строгая выпуклость – неравенству

$$f(x+y) > f(x) + (\nabla f(x),y), \quad ||y|| \neq 0, \tag{3.3.4}$$

а сильная выпуклость – неравенству

$$f(x+y) \geq f(x) + (\nabla f(x),y) + \rho\, ||y||^2/2 \tag{3.3.5}$$

Из неравенств (3.3.3)-(3.3.5) вытекают следствия:

а) для выпуклой функции выполняется неравенство

$$(\nabla f(x) - \nabla f(y), x - y) \geq 0; \tag{3.3.6}$$

б) для строго выпуклой функции при $x \neq y$ - строгое неравенство (3.3.6);

в) для сильно выпуклой функции

$$(\nabla f(x) - \nabla f(x), x - y) \geq \rho ||x-y||^2 \tag{3.3.7}$$

Для дважды дифференцируемой функции выпуклость эквивалентна, условию $\nabla^2 f(x) \geq 0$, а сильная выпуклость – выполнению условия $\nabla^2 f(x) \geq \rho I$ для всех x.

Для дифференцируемой, сильно выпуклой функции $f(x)$ точка минимума x^* (как мы увидим ниже) существует, единственна и $\nabla f(x^*)=0$. Поэтому из неравенств (3.3.5), (12) получим

$$f(x) \geq f(x^*) + \rho\, ||x-x^*||^2/2 \quad , \tag{3.3.8}$$

$$(\nabla f(x), x-x^*) \geq \rho\, ||x-x^*||^2 \quad , \tag{3.3.8}$$

$$||\nabla f(x)|| \geq \rho\, ||x-x^*|| \; . \tag{3.3.9}$$

Для сильновыпуклой функции также справедлива оценка

$$||\nabla f(x)||^2 \geq 2\rho\, (f(x) - f(x^*)) \tag{3.3.10 }$$

Согласно (3.3.10), возможное полное уменьшение функции в результате минимизации ограничено сверху.

Для дифференцируемой функции, градиент которой удовлетворяет условию Липшица

$$||\nabla f(x) - \nabla f(y)|| \leq L||x-y|| \tag{3.3.11}$$

и $f(x) \geq f^*$, $\forall x$, справедлива оценка

$$||\nabla f(x)||^2 \leq 2L\, (f(x) - f^*). \tag{3.3.12}$$

Если функция выпукла и дифференцируема, а ее градиент удовлетворяет условию Липшица, тогда

$$(\nabla f(x) - \nabla f(y), x - y) \geq L^{-1} \|\nabla f(x) - \nabla f(y)\|. \qquad (3.3.13)$$

3.3.1. Условия экстремума задачи безусловной минимизации

Условия экстремума являются основой, на которой строятся методы решения задач оптимизации и дают информацию о свойствах решения. Доказательства условий экстремума и их вид часто указывают путь построение методов оптимизации. В этом разделе будут рассмотрены условия экстремума задачи без ограничений.

Точка x называется стационарной, если в ней выполнено условие

$$\nabla f(x) = 0 \qquad (3.3.14)$$

Теорема 1. (Необходимое условие 1 порядка). Пусть x^*- точка минимума f(x), $x \in R^n$ и f(x) дифференцируема в x^*, тогда выполняется условие стационарности (3.3.14).

Доказательство следует из возможности линейного представления функции в точке x^*. Не всякая из точек, удовлетворяющих (20) является точкой минимума. Она может быть точкой максимума или седловой точкой.

Теорема 2. (Достаточное условие 1-го порядка). Пусть f(x) - выпуклая функция, дифференцируется в точке x^* и выполняется условие (20). Тогда x^* - точка глобального минимума f(x) на R^n.

Доказательство следует из (3.3.3).

Теорема 3. (Необходимое условие 2-го порядка). Пусть x^* - точка минимума f(x), $x \in R^n$ и f(x) дважды дифференцируется в x^*. Тогда $\nabla^2 f(x^*) \geq 0$.

Теорема 4. (Достаточное условие 2-го порядка). Пусть в точке x^* f(x) дважды дифференцируема, выполнено условие (3.3.14) и $\nabla^2 f(x^*) > 0$, тогда x_y- точка локального минимума.

Циклический покоординатный поиск в задачах безусловной Рассматривается задача $min\ \varphi(x)$ при $x \in E_n$. Идея метода простого покоординатного поиска заключается в следующем. Берется начальная точка x_0 и вектор λ величина шагов по каждому направлению, и полагаем $x^*= x_0$. Из точки x^* пытаемся перейти в точку x' изменением только первой координаты на величину $\lambda[1]$. Если эта попытка удачная, т.е. $\varphi(x_0) < \varphi(x')$, то увеличиваем величину шага $\lambda[1]$, $\lambda[1]:= \lambda\cdot\alpha$ и переходим в точку x', и полагая $x^* = x'$. Если попытка неудачная , т.е. $\varphi(x')\geq\varphi(x^*)$, то изменяем величину и направление шага $\lambda[1]$, $\lambda[1] := \lambda[1] \cdot \beta$, где $-1<\beta<0$. Затем переходим к следующему координатному направлению. По окончании спуска по всем n координатам оцениваем $| x_0 - x^*|$ и $|\lambda|$. Если эта величина меньше заданного ε, то процесс прекращаем, иначе полагаем $x_0 = x_0$ и процесс начинаем заново. Заметим, что возможен такой случай, $x_0 = x^*$, а $|\lambda|>\varepsilon$. Это может произойти тогда, когда величины $\lambda[i]$, $i=1, .. n$ завышены и все шаги неудачные.

Алгоритм простого покоординатного поиска

Начало 1

Массивы s, x_0, x^*, λ[1:n]

F x_0 :=φ(x_0);

x^*= x_0 ;

F x^* := F x_0

M: Для k := 1 до n выполнять

Начало 2

s:= e_k; {здесь и далее e_k - k – ый орт}

x' := x^* +λ [k]* s;

F x':=φ(x');

Если F x' < F x^* то

Начало 3 x^* = x' ;

$Fx* := F x'$;

$\lambda[k] := \lambda[k]*\alpha$

Конец 3

Иначе $\lambda[k] := \lambda[k]*\beta$

Конец 2

Если $(|x_0 - x^*| > \varepsilon)$ и $(|\lambda| \geq \varepsilon)$ то

Начало 4

$x_0 = x^*$;

$Fx_0 = Fx^*$

Идти M

Конец 4

Конец 1.

Алгоритм исчерпывающего покоординатного поиска

В отличие от простого покоординатного поиска, в исчерпывающем покоординатном поиске при удачном переходе x * в x ‘ предполагается дальнейший спуск по этому направлению до тех пор , пока не получим неудачный переход. В этом случае осуществляется переход к следующему направлению спуска. Отличие алгоритма исчерпывающего покоординатного поиска от предыдущего алгоритма заключается лишь в блоке 2

Начало 2

$s := e_k$; {здесь и далее e_k - k – ый орт}

$x' := x^* + \lambda[k]*s$;

$F x' := \varphi(x')$;

Пока F x ‘ < F x * выполнить

Начало 3

$x^* = x'$;

$Fx^* := F x'$;

$\lambda[k]:= \lambda [k]*\alpha$

$x' := x^* +\lambda [k]* s$;

$F x' :=\varphi(x')$;

Конец 3

$\lambda[k]:= \lambda[k]*\beta$

Конец 2

3.4.3.Метод экстремального покоординатного поиска

Этот метод заключается в последовательном отыскании минимума по всем координатным направлениям пространства E_n . По каждому из этих направлений функция $\varphi(x')$является функцией одной переменной, для которой методы минимизации известны. Если текущее значение переменной *x* равно x^* , то точка x' находится из условия :

$\varphi (x')=min\ \varphi(x^*+s\cdot\lambda [k]* s)$, где $s:= e_k$.

В этом случае блок *Начало 2…Конец2* будет иметь вид

Начало 2

$s:= e_k$;

$\lambda'[k]:=\text{argmin}\ \varphi(x^*+ \lambda * s)$

$x' := x^* +$ [k]* s; $x^* = x'$;

Конец 2

При этом условие $|\lambda | >\varepsilon$ можно не проверять, так как оно эквивалентно условию $|x_0- x^*| >\varepsilon$

3.5.Метод наискорейшего спуска

Рассмтрим метод минимизации функции $f(x)$, где $x \in E_n$ из заданного начального приближения x_0 по итерационной формуле

$$x_{k+1}=x_k+\lambda_k*S_{k,},\ k=0,1,2, \quad (3.5.1)$$

где x_k – приближение к решению задачи на k-ой итерации;

$S_{k,}$- направление убывания функции $f(x);$

λ_k – длина шага вдоль направления S_k .

Итак, пусть $f(x)=f(x_1,x_1,...x_n)$ дифференцируемая функция, заданная на E_n , а $x^{(k)}=(x_1^{(k)},x_2^{(k)},...,x_n^{(k)})$ некоторая текущая точка.

Выберем начальное приближение $x^{(0)}$. Общих рекомендаций для выбора начального приближения $x^{(0)}$ не существует.

Для ускорения решения задачи следует выбирать начальное приближение достаточно близко от искомой оптимальной точки x^*.

Известно, что направление , вычисленное в точке x, дает направление наискорейшего возрастания функции, а направление $\nabla f(x)=- f'(x)$ дает направление наискорейшего убывания функции

Если $x^{(k)}$ нестационарная точка (т. е. $\nabla f(x^{(k)})<0$), то при движении в направлении $\nabla f(x^{(k)})$ функция $f(x)$ на некотором промежутке обязательно будет убывать, поэтому возникает идея такого выбора шага, чтобы движение в указанном направлении продолжалось до тех пор, пока убывание не прекратится.

Выразим зависимость значения $f(x)$ от λ ($\lambda > 0$). полагая, что

$$x = x^{(k)} + \lambda \nabla f(x^{(k)}) \quad , \quad (3.5.2)$$

тогда

$$f(x)=f(x^k+\lambda\nabla^{)f}(x^{(k)})) \quad (3.5.3)$$

Так как точка $x^{(k)}$ известна, то функция зависит только от λ, т. е будем иметь $f(x)=f(\lambda)$, или в покоординатной форме:

$$f(\lambda)= f(x^k{}_1+\lambda\partial f(x^{(k)})/\partial x_1, \ldots,)\ x^k{}_n+\lambda\partial f(x^{(k)})/\partial x_n) \quad (3.5.4)$$

Чтобы добиться наименьшего значения *f*, нужно выбрать такое значение λ', которое минимизирует функцию *f(λ), т.е* $f(\lambda') = \min(f(\lambda))$. Для вычисления λ' используется необходимое условие существования экстремума $df(\lambda)/d\lambda = 0$. Если для любого $\lambda>0$ функция *f(x)* не ограничена снизу, то она не имеет минимума. В противном случае, на основе (3) получаем

$$\partial f/\partial\lambda = (\partial f(x)/\partial x_1)\times(\partial x_1/\partial \lambda), \ldots, (\partial f(x)/\partial x_n)\times(\partial x_n/\partial \lambda) \quad (3.5.5)$$

откуда следует

$$\partial f/\partial\lambda = (\partial f(x)/\partial x_1)\times(\partial f(x^{(k)})/\partial x_1) + \ldots + (\partial f(x)/\partial x_n)\times(\partial f(x^{(k)})/\partial x_n) = \nabla f(x)\nabla f(x^{(k)}) \quad (3.5.6)$$

Если следующая точка $x^{(k+1)}$ соответствует оптимальному значению $\lambda=\lambda'$, то в ней должно выполняться условие $df(\lambda')/d\lambda = 0$, и λследует находить из условия $\nabla f(x^{(k+1)})\nabla f(x^{(k)})=0$ или

$$\nabla f(x^k + \lambda'\nabla f(x^{(k)}))\nabla f(x^{(k)})=0 \quad (3.5.7)$$

Условие (3.5.7) означает равенство нулю скалярного произведения градиентов функции *f* точках $x^{(k+1)}$ и $x^{(k)}$. Геометрически оно означает, что векторы градиентов функции *f* в указанных точках перпендикулярны. Таким образом, $\nabla f(x^{(k+1)})$ перпендикулярен линии уровня, проходящей через данную точку и ,следовательно, вектор $\nabla f(x^{(k)})$ является касательным к этой линии. Движение в направлении градиента $\nabla f(x^{(k)})$ следует продолжать до тех пор, пока он пересекает линии уровня оптимизируемой функции. После того как точка $x^{(k+1)}$ найдена, она становится текущей для очередной итерации. Признаком достижения стационарной точки служит достаточно малое изменение координат точек, рассматриваемых на последовательных итерациях и приближение координат вектора $\Delta f(x^{(k)})$ к нулю.

Пример . Определить градиентным методом минимум функции $z=x_1^2 + x_2^2- 4x_1 - 2x_2 -5$, начав итерационный процесс из точки X^0= (0,0).

<u>Решение:</u> Находим $\partial f/\partial x_1=-4+2x_1$, $\partial f/\partial x_2=-2+2x_2$, тогда

$\nabla f^0 = -(\partial f/\partial x_1, \partial f/\partial x_2) = (4-2x_1, 2-2x_2) = (4-2*0, 2-2*0) = (4,2)$.

Итерация 1.

Начальная точка $X^0=(0, 0)$. Шаг по направлению (по градиенту) $\nabla f^0 = (4, 2)$.

Вычислим очередное приближение по формуле X' =(0, 0)+$\lambda\nabla$ f ^{0} = (0+4*λ, 0+2*λ)=(4λ, 2λ)

∇ f $'$ = (4-2x_1',2-2x_2')=(4-2*4λ), 2-2*2λ)) = (4-8λ, 2-4λ)

Используя необходимое условие экстремума, имеем:

$$\frac{d\Delta z}{d\lambda}=\left(\frac{\partial f}{\partial x_1}\right)'\left(\frac{\partial f}{\partial x_1}\right)^0+\cdots+\left(\frac{\partial f}{\partial x_n}\right)'\left(\frac{\partial f}{\partial x_n}\right)^0=(\nabla f)'*(\nabla f)^0=0$$

Подставим $\nabla f\,'$ и $\nabla f\,^0$, получим

$(\nabla$ f $')$ * $(\nabla$ f $^0)$ = (4-8λ)*4 + (2-4λ) *2= 16 - 32λ + 4 - 8λ= 20 - 40λ = 0

Откуда λ^0 = 0,5, при этом $\partial^2\Delta z/\partial^2\lambda = -40 < 0$, следовательно, найденное значение λ является точкой минимума Δz. С помощью величины λ^0 получаем новую точку *X'=(0+4*0,5,0+2-0,5)=(2, 1)*

Итерация 2.

Начальная точка *X'=(2, 1)* ;

Вычислим:

∇ f $'$ = (4-2x_1', 2-2x_2')=(4-2*2), 2-2*1) = (0,0)

Шаг по направлению равен: $\nabla f\,'$ *= (0,0), поэтому*

Следовательно, *X'=(2, 1)* является стационарной точкой и дальнейшее перемещение вдоль градиента невозможно. Так как функция *z* выпуклая, то в найденной точке достигается глобальный минимум Zmin = $2^2 + 1^2 - 4*2 - 2*1 - 5 = -10$.

Таким образом, x*=(2, 1), Z*=-10.

Алгоритм метода наискорейшего спуска

Начало1

x:=x0;

PRF_x:= f'(x);

Пока ||f'(x) || > ε выполнить

Начало2

S=- f '(x);

λ:=argmin φ(x+□ s), здесь λ ≥0

x:= x+□ s

к

Конец 2

Конец 1

4. Лабораторный практикум.

4.1.Лабораторная работа № 1.

Поиск интервала, содержащего точку минимума.

Большинство методов поиска минимума функции $\varphi(x)$ одной переменной основаны на построении последовательности стягивающихся отрезков (промежутков), содержащих точку минимума. Начальный промежуток, как правило, неизвестен. Алгоритм поиска начального промежутка для унимодальной функции основан на поиске точек $a<x<b$, в которых $f(x) \le min(f(a), f(b))$. *Т*огда точка x^* минимума $\varphi(x)$ содержится в промежутке $[a,b]$

Правило исключения интервалов.

Пусть функция $f(x)$ - унимодальна на замкнутом интервале $a < x < b$, а её минимум достигается в точке x^*. Рассмотрим точки x_1 и x_2, расположенные в интервале следующим образом $a < x_1 < x_2 < b$.

Сравнивая значения функции в точках x_1 и x_2, можно сделать следующие выводы:

1) Если $f(x_1) > f(x_2)$, то точка минимума не лежит в интервале (a, x_1), т. е. $x^* \in (x_1, b)$.

Если $f(x_1) < f(x_2)$, то точка минимума не лежит в интервале (x_2, b), т.е. $x^* \in (a, x_2)$

Если $f(x_1) = f(x_2)$, то точка минимума не лежит в интервале (a, x_1) и (x_2, b), т.е. $x^* \in (x_1, x_2)$

Достоинство:

1) Правило исключения интервалов устраняет необходимость перебора всех точек.

2) Не требуется выполнение условия дифференцируемости функций, и даже нет необходимости записи функции в математическом виде. Единственное требование: возможность определения значения функции в заданных точках с помощью расчетов или в результате экспериментов.

Методы поиска, основанные на использовании правила исключения интервалов, состоят из 2 этапов:

1) Этап установления границ интервала, в котором сходится точка минимума (достаточно широкого интервала).

2) Этап уменьшения интервала до заранее установленной величины.

Критерии сходимости:

1) Длина интервала меньше заданного значения $| x_1 - x_2| < \varepsilon$

2) Разность между двумя оценками значений функции меньше заданной погрешности $|f(x^k) - f(x^{k+1})| < \varepsilon_1$.

Этот поиск реализует следующий алгоритм.

Алгоритм поиска промежутка локализации минимума функции φ(x)

Задаются произвольные значения x и Δx, где x - начальная точк.; Δx. - начальный шаг, m - коэффициент увеличения шага.

НАЧАЛО 1

Fx:=$\varphi(x)$;

y:= $x + \Delta x$;

Fy :=$\varphi(y)$;

Если Fy>Fx То

Начало 2 {меняем x и y местами}

z=x; Fz=Fx;

x=y; Fx=Fy;

y:=z; Fy=Fz;

Конец 2

Пока Fx< Fx выполнять

Начало 3

x:=y; Fx:=φ(x);

Δx:=m Δx

y:=x+ Δx

Fy:= φ(y);

Конец 3

Если $\Delta x > 0$, то

Начало4

a:=x- Δx/m

b:=y

Конец 4

Иначе

Начало5

a:=y

b:=x- Δx/m

Конец5

Конец1.

В результате работы алгоритма подучаем интервал *[a,b]*, содержащий точку оптимума и точку $x \in [a,b]$, в. которой известно значение $Fx = \varphi(x)$;

Задание

Определить промежуток локализации минимума функции

$\varphi(x) = (x\text{-}d)\,(x\text{-}l)^3$. В качестве начальных значений взять x=0, Δx =1, m =2

Варианты заданий

Вариант	1	2	3	4	5	6	7	8	9	10	11	12	13
d	5	7	8	8	1	3	3	1	2	-5	-7	-8	-8
l	6	8	9	10	4	5	4	3	4	-6	-8	-9	10
Вариант	14	15	16	17	18	19	20	21	22	23	24	25	
d	-1	-3	-3	-1	-2	0	0	2	-2	-4	3	4	
l	-4	-5	-4	-3	-4	-5	5	6	-6	0	6	9	

Результаты расчетов поместить в таблицу, имеющую вид:

№ шага k	x	Fx	*Δ*x	y	Fy

Под номером k здесь понимается порядковый номер входа в блок Начало 3 Конец 3.

Значения a и b получены в результате алгоритма выписываются ниже таблицы в форме ответа..

4.2. Лабораторная работа № 2 . Методы нулевого порядка.

Методы оптимизации без ограничений называются методами безусловной оптимизации.

Классификация методов безусловной оптимизации:

1) Методы 0-го порядка (прямые методы) - методы, использующие только значения функций.
2) Методы 1-го порядка – методы, использующие первые производные.
3) Методы 2-го порядка - методы, использующие первые и вторые производные.

Производные могут вычисляться аналитически или численно. Методы 3) при наименьшем числе шагов приводят к точкам, достаточно близким к точкам минимума. Это не означает, что они являются наиболее эффективными. Иногда *f(x)* представляет собой настолько сложную функцию, что 1-я и 2-я производные не могут быть получены аналитически, а их численные вычисления оказываются очень грубыми. Кроме того, вычисление этих производных может потребовать больше машинного времени, чем вычисление значений функций в необходимом для отыскания минимума числе точек.

Таким образом, невозможно выделить какой-либо метод, который был бы наиболее эффективным для решения любой задачи.

Фактически все одномерные методы поиска, используемые на практике, основаны на предположении, что исследуемая функция в допустимой области унимодальна. Для унимодальной функции *f(x)* сравнение значений функции *f(x)* в двух различных точках интервала поиска позволяет определить: в каком именно, из заданных двумя указанными точками подинтервалов, точка оптимума отсутствует.

Поиск оптимума заключается в последовательном исключении подинтервалов и, следовательно, уменьшении интервала поиска.

Задание

Определить минимум функции $\varphi(x) = (x\text{-}d)\ (x\text{-}l)^3$ методом симметричного и несимметричного поиска. . В качестве начальных значений взять x=0, Δx =1, m =2 Начальный промежуток определить методом,рассмотренным в лабораторной работе 2.

Варианты заданий

Вариант	1	2	3	4	5	6	7	8	9	10	11	12	13
d	5	7	8	8	1	3	3	1	2	-5	-7	-8	-8
l	3	6	7	5	2	9	4	1	9	7	5	3	4
Вариант	14	15	16	17	18	19	20	21	22	23	24	25	
d	2	3	5	4	6	7	8		4	9	7	5	
l	-4	-5	-4	-3	-4	-5	5	6	-6	0	6	9	

Результаты расчетов поместить в таблицу, вид придумать самостоятельно

4.3.Лабораторная работа № 3.
Методы первого порядка.

Методы деления отрезка пополам и метод касательных

Рассматриваются унимодальные дифференцируемые функции $\varphi(x), x\in[a,b]$

Пусть в точке y$\in[a,b]$ найдено значение $\varphi'(x)$. Если $\varphi'(x)<0$ то на отрезке $[a,y]$ минимума функции нет, и наоборот, если , $\varphi'(x)>0$ то на отрезке $[y,b]$ минимума нет. В первом случае следует $a:=y$, во втором $b:=y$. Метод деления пополам и метод касательных различают между собой способом выбора пробной точки y. В методе деления пополам y находится в середине отрезка $[a,b]$, в методе касательных y является абсциссой пересечения касательных, проведенных к *φ(x)*, в точках *φ(x), a* и *b*. В начале работы алгоритма целесообразно произвести проверку знака производной в концевых точках. Это позволит в ряде случаев сразу выявит оптимальные значениями.

Метод касательных

Метод касательных, являющийся комбинацией метода Ньютона и общей схемы исключения интервалов, ориентирован на нахождение корня уравнения *fr(x)* в интервале *(a, b)*, если, разумеется, такой корень существует.

Предположим, что в процессе поиска стационарной точки функции *f(x)* в интервале *(a, b)* обнаружены две точки *L* и *R*, в которых знаки производной различны. В этом случае алгоритм метода касательных позволяет аппроксимировать функцию *fr(x)* прямой линией, соединяющей две точки, и найти точку, в которой касательная к графику *f(x)* пересекает ось абсцисс .Тогда следующее приближение к стационарной точке x^* определяется по формуле

$$z = R - \frac{f'(R)}{[f'(R) - f'(L)]/(R - L)}.$$

Задание

Определить минимальное значение функции $\varphi\ (x) = (x\text{-}d)(x\text{-}l)^3$ методом деления отрезка пополам (методом Больцано). Значение e = 0.05. Начальный интервал вычисляется из первой лабораторной. Варианты заданий взять из лабораторной работы № 2. Результаты вычислений представить в виде таблицы.

4.4. Лабораторная работа № 4

Методы второго порядка. Методы Ньютона.

Пусть найден некоторый промежуток *[a,b]* локализации точки минимума функции *φ (x)*. Полагаем, что на *[a,b]* 0< m≤ *φ'' (x)* ≤M. Пусть ьакже y пробная точка этого отрезка. В точке y разложим *φ (x)* в ряд Тэйлора и в качестве квадратичной аппроксимации возьмем 3 член этого ряда

$$\varphi(x)=\varphi(y)+\varphi'(y)*(x-y)+1/2*\varphi''(y)(x-y)^2$$

Минимум этого полинома достигается в точке

$x=y-\varphi'(y)/\varphi''(y)$.

Её и примем в качестве нового приближения к экстремуму x* функции *φ (x)*. Для получения более точного решения полагают y=x и определяют новое приближение. В этом и состоит метод Ньютона (метод квадратичной аппроксимации). Заметим, что если отрезок *[a,b]* локализует точку максимума, то значение $x = y - \varphi'(y)/\varphi''(y)$ будет приближенным значением максимума функции *φ (x)*. Этот метод быстро сходится, однако существуют случаи, когда метод расходится. Признаки расходимости является выполнение условия $x \in [a,b]$.

Задание.

Определить минимум функции $\varphi(x) = (x-d)(x-l)^3$ методом Ньютона. . В качестве начальных значений взять x=0, Δx =1, m =2

Начальный промежуток определить методом,рассмотренным в лабораторной работе 2.

Варианты заданий

Вариант	1	2	3	4	5	6	7	8	9	10	11	12	13
d	4	3	5	6	2	3	4	7	8	9	1	3	5
l	3	6	7	5	2	9	4	1	9	7	5	3	4
Вариант	14	15	16	17	18	19	20	21	22	23	24	25	
d	4	6	8	9	2	4	6	7	3	4	5	7	
l	-4	-5	-4	-3	-4	-5	5	6	-6	0	6	9	

Результаты расчетов поместить в таблицу, вид придумать самостоятельно

4.5. Лабораторная работа № 5

Циклический покоординатный поиск в задачах безусловной оптимизации

Рассматривается задача *min* $\varphi(x)$ при $x \in E_n$. Методы циклического покоординатного поиска рассмотрены главе5.

Задание

Определить минимум функции $\varphi(x) = (\xi_1 - d_1)^2/4 + (\xi_2 - d_2)^2/9 + d_3$

За начальное приближение взять точку $x_0=(0, 0)$, $\lambda=(2,2)$.

Варианты задания d_1, d_2 и d_3 представлены в следующей таблице.

Таблица значений

№ варианта	d_1	d_2	d_3
1	1	2	1
2	3	4	2
3	1	1	1
4	1	2	2
5	5	6	1
6	2	4	1
7	1	3	2
8	1	5	3
9	2,5	2	3
10	3	3	1
11	1	1	3
12	3	5	1
13	3	6	2
14	3	3	3
15	3	4	2
16	2,5	1	3
17	2,7	7	4
18	4	6	2

19	1	3	4
20	0,5	4	2
21	1	5	4
22	3	2	1
23	4	1	4
24	1	1	1
25	4	2	1

4.6.Лабораторная работа № 6

Метод наискорейшего спуска

. В данной лабораторной работе требуется найти минимум функции *$\varphi(x)$, используя метод наискорейшего спуска.* Теоретические основы и алгоритм метода рассмотрены в в главе 3.

Задание

Определить минимум функции $\varphi\ (x) = (\xi_1 - d_1)^2/4 + (\xi_2 - d_2)^2/9 + d_3$

Варианты задания d_1,d_2 и d_3 представлены в таблице лабораторной работы №5. Значение ε взять равным 0,01. За начальное приближение взять точку x_0=(0, 0).

Литература

1 Тетерев А.Г. Методы одномерной оптимизации. Методические указания. КуГУ, г.Куйбышев, 1983 г.

2 Беллман Р., Дрейфус С. Прикладные задачи динамического программирования. М., 1965

3 Васильев Ф.П. Численные методы решения экстремальных задач. М., 1980.

4 Карманов В.Г. Математическое программирование. М.,1980.

5 Моисеев Н.Н., Иванилов Ю.П., Столярова Е.М. Методы оптимизации. М. 1978.

6 Стронгин Р.Г. Численные методы в многоэкстремальных задач. М., 1978

7 Сухарев А.Г. Оптимальный поиск экстремума. Изд. МГУ, 1975.

8 Тетерев А. Г. Методы возможных направлений в математическом программировании. Куйбышев, 1980.

9 Тетерев А.Г. Анализ сходимости и устойчивости методов одномерной оптимизации. Вестник МГУ, серия XV, вып. 4.,1981

10 Уайлд Д.Дж. Методы поиска экстремума. М., 1967.

11 Химмельблау Д. Прикладное нелинейное программирование. М., 1975.

Приложение1.**Образец оформления отчета по лабораторным работам по методам оптимизацию**

Отчет по лабораторной работе №5 «методы второго порядка. Методы Ньютона, ДСК, Пауэла»

Меньшаковой А.С. гр 12401

Вариант 8

Задание: 1. Для функции $\varphi(x) = (x-1)(x-3)^3$ выполнить поиск минимума функции $\varphi(x)$ методом Ньютона для e=0,005.

2. Для функции $\varphi(x) = (x-1)(x-3)^3$ выполнить поиск минимума функции $\varphi(x)$ методом ДСК для e=0,0005.

3. Для функции $\varphi(x) = (x-1)(x-3)^3$ выполнить поиск минимума функции $\varphi(x)$ методом Пауэла для e=0,005.

1.метод Ньютона

a=1 b=7

x	Fx	y	Fy
2,61188	-0,09424	2,68	-0,05505
2,61188	-0,09424	2,61188	-0,09424
2,51155	-0,17615	2,61188	-0,09424
2,51155	-0,17615	2,51155	-0,17615
2,35249	-0,36718	2,51155	-0,17615
2,35249	-0,36718	2,35249	-0,36718
2,07341	-0,85394	2,35249	-0,36718
2,07341	-0,85394	2,07341	-0,85394
1,50896	-1,68714	2,07341	-0,85394
1,50896	-1,68714	1,50896	-1,68714

x=1,50896   Fx=-1,68714

2.метод ДСК

a=1   b=7

Fa=0 Fb=384

x	Fx	y	Fy
2,28975	-0,4621	5,71	93,74083
1,51657	-1,68628	2,28975	-0,4621

x=1,51657   -1,68628

3.метод Пауэла

a=1   b=7

Fa=0 Fb=384

x	Fx	y	Fy
1,45274	-1,67702	1,75	-1,46484
1,54078	-1,68028	1,45274	-1,67702
1,50192	-1,68748	1,54078	-1,68028

x=1,50086   Fx=-1,6875

Методические материалы

Власова Ираида Абрамовна

Методы одномерной оптимизации

Методические указания