НАО «Восточно-Казахстанский технический университет имени Д. Серикбаева»

Дисциплина «Методы оптимизации»

ЛЕКЦИЯ 8

Методы золотого сечения и Фибоначчи

Образовательная программа 6B06103 Математическое и компьютерное моделирование

Объём: 2 академических часа. **Место в рабочей программе:** лекция 8.

Цель: научиться минимизировать унимодальную функцию одной переменной методами золотого сечения и Фибоначчи, обосновывать точность результата и сравнивать затраты на вычисление функции.

План занятия: принцип исключения интервалов; вывод пропорции золотого сечения; повторное использование значений функции; сходимость и вычислительные затраты; числа Фибоначчи и планирование поиска; заключительное смещение; подробный пример; программы и самостоятельные задания.

Результаты обучения: студент строит пробные точки обоих методов, выполняет итерации вручную, определяет число вычислений, различает допуск по координате и по функции, проверяет результат по конечному отрезку и объясняет особенности последнего шага метода Фибоначчи.

1 Постановка задачи и правило исключения

Рассмотрим задачу

$$f(x) \to \min, \qquad x \in [a_0; b_0].$$

Предположим, что функция непрерывна на заданном отрезке и строго унимодальна: до единственной точки минимума x^* она строго убывает, а после неё строго возрастает. Допускается минимум на границе; тогда одна из монотонных частей отсутствует. Непрерывность обеспечивает существование минимума на отрезке, а строгая унимодальность позволяет исключать его части по двум значениям функции [1].

Выберем внутренние точки $a < y < z < b$. Если $f(y) < f(z)$, минимум не может находиться правее z. Иначе обе пробные точки лежали бы на убывающей части и выполнялось бы противоположное неравенство. Поэтому сохраняем $[a; z]$. Аналогично, при $f(y) > f(z)$ сохраняем $[y; b]$.

При точном равенстве $f(y) = f(z)$ строгая унимодальность даёт $x^* \in [y; z]$. Для единообразия теоретических алгоритмов разрешается сохранить более длинный отрезок $[a; z]$ и использовать ветвь $f(y) \le f(z)$. Тогда формулы стандартного сокращения остаются неизменными. Отдельное правило для равенства может ускорить поиск, но изменяет его заранее рассчитанную схему.

Вычислительное замечание. Равенство чисел типа float не доказывает точного равенства математических значений. В приложенных программах при совпадении вычисленных значений поиск прекращается со специальным статусом и сохраняет текущий отрезок. Такой результат нельзя автоматически считать достигшим заданной точности.

Оба изучаемых метода относятся к методам нулевого порядка: для шага достаточно значений функции, производные не используются. Производная остаётся полезной для предварительного обоснования унимодальности и проверки учебного примера.

2 Почему выгодно сохранять пробную точку

В дихотомии на каждой итерации обычно вычисляются два новых значения функции. Если одно вычисление включает решение дифференциального уравнения, проведение моделирования или обработку большого массива данных, именно эти обращения определяют основную стоимость поиска.

Пусть после сравнения сохранён отрезок $[a; z]$. Старая точка y лежит внутри него, и значение $f(y)$ уже известно. Желательно использовать её как одну из двух точек следующего сравнения. Тогда потребуется вычислить только одну новую точку. При сохранении правой части аналогично используется старая точка z.

Повторное использование означает сохранение **обоих** объектов: координаты и значения функции. Перенести координату, а затем заново вычислить значение — значит получить правильный геометрический шаг, но потерять вычислительное преимущество метода.

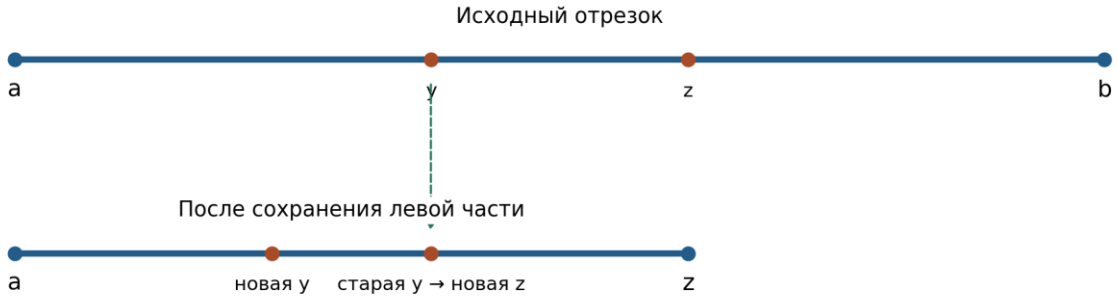


Рисунок 1 — Старая левая точка становится правой точкой нового отрезка

Метод золотого сечения сохраняет одну и ту же пропорцию деления на всех шагах. Метод Фибоначчи меняет пропорцию в зависимости от числа оставшихся сравнений. В обоих случаях новые точки выбираются так, чтобы использовать предыдущее вычисление.

3 Вывод золотой пропорции

Обозначим длину текущего отрезка через $L = b - a$. Будем располагать точки симметрично:

$$y = a + (1 - \tau)L, \qquad z = a + \tau L, \qquad \frac{1}{2} < \tau < 1.$$

При сохранении левой части её длина равна τL. Старая точка y должна стать правой пробной точкой этого нового отрезка. По неизменной пропорции её расстояние от a должно равняться $\tau \cdot \tau L$. Но до сокращения это расстояние было $(1 - \tau)L$. Следовательно,

$$(1 - \tau)L = \tau^2 L, \qquad \tau^2 + \tau - 1 = 0.$$

Положительный корень, принадлежащий нужному интервалу, равен

$$\tau = \frac{\sqrt{5}-1}{2} \approx 0{,}6180339887, \qquad 1-\tau = \tau^2 \approx 0{,}3819660113.$$

Величина $\varphi = 1/\tau = \left(1+\sqrt{5}\right)/2$ называется золотым отношением. В алгоритме удобнее использовать τ, поскольку оно непосредственно выражает долю сохраняемой длины.

Проверка правой ветви. Если сохраняется $[y; b]$, его длина также равна τL. Старая точка z становится левой пробной точкой. Её новое относительное положение равно

$$\frac{z-y}{b-y} = \frac{2\tau-1}{\tau} = 1-\tau.$$

Последнее равенство следует из $\tau^2 = 1-\tau$. Значит, одна и та же пропорция обеспечивает повторное использование точки в обеих ветвях.

В ручном решении значения можно показывать с 5–6 знаками, однако в вычислениях следует сохранять больше разрядов. Замена τ на 0,618 уже в программе постепенно нарушает точное соответствие между старой и новой точками.

4 Алгоритм золотого сечения

Сначала задают $a_0 < b_0$ и допуск по координате $\varepsilon_x > 0$. Если исходный отрезок уже удовлетворяет $b_0 - a_0 \le 2\varepsilon_x$, его середину можно сразу принять за приближение. В противном случае вычисляют две пробные точки и два значения функции.

Пусть перед очередным сравнением известны $a, b, y, z, f_y = f(y), f_z = f(z)$. Используем следующие переходы.

Левая ветвь при $f_y \le f_z$. Сохраняем $[a; z]$, а старые y, f_y используем как новую правую точку и её значение:

$$b_{\text{new}} = z, \qquad z_{\text{new}} = y, \qquad f_{z,\text{new}} = f_y.$$

Если длина ещё велика, строим только новую левую точку:

$$y_{\text{new}} = a + (1-\tau)(b_{\text{new}} - a).$$

Правая ветвь при $f_y > f_z$. Сохраняем $[y; b]$, переносим старые z, f_z в новую левую точку:

$$a_{\text{new}} = y, \qquad y_{\text{new}} = z, \qquad f_{y,\text{new}} = f_z.$$

Для продолжения поиска вычисляем

$$z_{\text{new}} = a_{\text{new}} + \tau(b - a_{\text{new}}).$$

В обеих ветвях после изменения границ сначала проверяют условие остановки и лишь затем вычисляют новое значение функции. Если необходимая точность уже достигнута, дополнительное пробное вычисление не требуется.

Критерий остановки и результат:

$$b_k - a_k \le 2\varepsilon_x, \qquad \hat{x} = \frac{a_k + b_k}{2}.$$

Поскольку $x^* \in [a_k; b_k]$, имеем

$$|\hat{x} - x^*| \le \frac{b_k - a_k}{2} \le \varepsilon_x.$$

Середина обеспечивает эту оценку независимо от того, где внутри отрезка расположен минимум. Лучшая из ранее вычисленных пробных точек может иметь меньшее значение функции, но для неё оценка через половину длины в общем случае не следует.

5 Сходимость и число вычислений

Каждое стандартное сокращение умножает длину отрезка на τ. После k сокращений в точной арифметике

$$L_k = \tau^k L_0, \qquad L_0 = b_0 - a_0.$$

При $L_0 > 2\varepsilon_x$ требуется $\tau^k L_0 \le 2\varepsilon_x$. Логарифмируя и учитывая $\ln\tau < 0$, получаем

$$k \ge \frac{\ln(2\varepsilon_x / L_0)}{\ln\tau}, \qquad k_{\min} = \left\lceil \frac{\ln(2\varepsilon_x / L_0)}{\ln\tau} \right\rceil.$$

Смена знака неравенства при делении на отрицательный логарифм обязательна. Если исходный отрезок достаточно короток, принимают $k_{\min} = 0$ и отдельно используют начальную проверку.

Для $k \ge 1$ первая итерация требует двух поисковых вычислений, остальные $k - 1$ итераций — по одному. Поэтому

$$N_{\text{search}} = k + 1.$$

Если дополнительно вычисляется значение функции в конечной середине, общий счётчик равен

$$N_f = k + 2.$$

Это правило соответствует приложенной программе: конечное значение рассчитывается отдельным вызовом. Если программа использует кэш и середина случайно совпадает с ранее вычисленной точкой, счётчик может отличаться. При сравнении реализаций необходимо заранее согласовать, какие обращения учитываются.

Пример оценки. Пусть $L_0 = 2{,}5$, а $\varepsilon_x = 0{,}05$. Тогда $2\varepsilon_x / L_0 = 0{,}04$, минимальное целое число сокращений равно 7. Следовательно, нужны 8 поисковых вычислений и ещё одно вычисление в конечной середине.

Оценка относится к координате. Малое различие $|f(y) - f(z)|$ само по себе не даёт такого же допуска по x: около пологого минимума значения могут быть почти одинаковыми даже у достаточно далёких точек.

6 Подробный пример золотого сечения

Минимизируем функцию

$$f(x) = x^2 + \frac{2}{x}, \qquad x \in [0{,}5; 3], \qquad \varepsilon_x = 0{,}05.$$

Проверим условия: на рассматриваемом отрезке функция непрерывна, а

$$f'(x) = 2x - \frac{2}{x^2}, \qquad f''(x) = 2 + \frac{4}{x^3} > 0.$$

Функция строго выпукла. Производная обращается в нуль при $x = 1$, отрицательна слева от этой точки и положительна справа. Значит, функция строго унимодальна, точное решение $x^* = 1$, $f(x^*) = 3$.

Первое сравнение. При $L_0 = 2{,}5$ получаем

$$y_0 = 0{,}5 + (1 - \tau) \cdot 2{,}5 \approx 1{,}454915,$$

$$z_0 = 0{,}5 + \tau \cdot 2{,}5 \approx 2{,}045085.$$

Вычислим $f(y_0) \approx 3{,}491428$ и $f(z_0) \approx 5{,}160327$. Левая величина меньше; сохраняем $[0{,}5;\ 2{,}045085]$. Значение в старой точке y_0 больше не вычисляем.

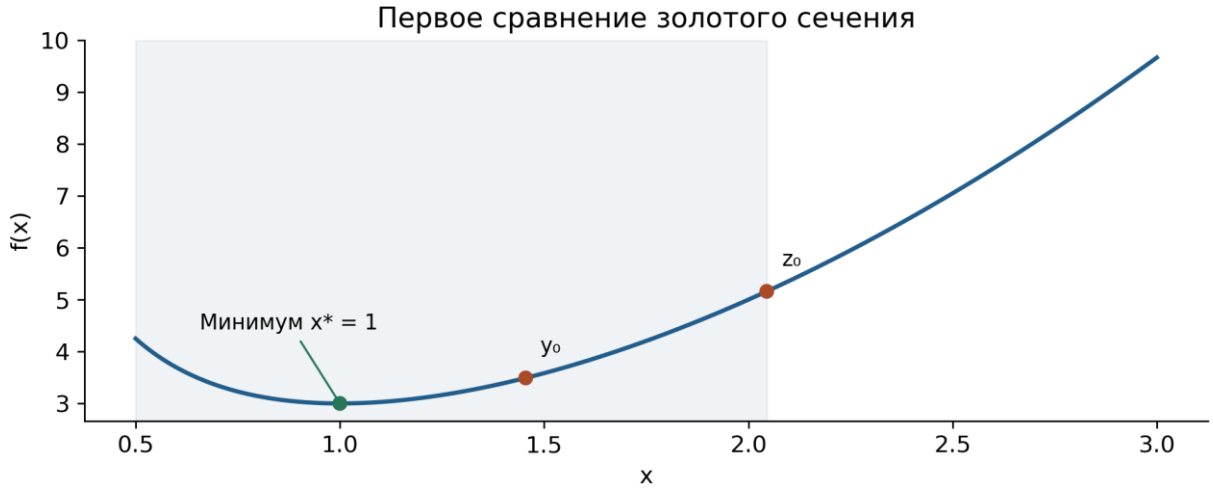


Рисунок 2 — Пробные точки и сохраняемая часть исходного отрезка

Второе сравнение. Старую точку y_0 используем как z_1. Новая левая точка

$$y_1 = 0{,}5 + (1 - \tau)(2{,}045084972 - 0{,}5) \approx 1{,}090170.$$

Её значение $f(y_1) \approx 3{,}023047$ меньше уже известного $f(z_1) \approx 3{,}491428$. Поэтому новый отрезок равен $[0{,}5;\ 1{,}454915]$. Для этой итерации потребовалось только одно новое вычисление.

Ниже показаны все сравнения. Границы в таблице округлены только для отображения; расчёт выполнялся с полной машинной точностью.

Шаг	y	z	$f(y)$	$f(z)$	Новый отрезок
1	1,45492	2,04508	3,491428	5,160327	[0,5; 2,04508]
2	1,09017	1,45492	3,023047	3,491428	[0,5; 1,45492]
3	0,86475	1,09017	3,060604	3,023047	[0,86475; 1,45492]
4	1,09017	1,22949	3,023047	3,138337	[0,86475; 1,22949]
5	1,00407	1,09017	3,000049	3,023047	[0,86475; 1,09017]
6	0,95085	1,00407	3,007497	3,000049	[0,95085; 1,09017]
7	1,00407	1,03695	3,000049	3,004	[0,95085; 1,03695]

После седьмого сокращения

$$[a_7; b_7] \approx [0{,}950849719; 1{,}036954353], \qquad L_7 \approx 0{,}086104634.$$

Следовательно,

$$\hat{x} \approx 0{,}993902036, \qquad \frac{L_7}{2} \approx 0{,}043052317 < 0{,}05.$$

Получено $f(\hat{x}) \approx 3{,}000112012$. Фактическая ошибка по координате равна приблизительно 0,006097964 и меньше гарантированной оценки. Всего выполнено 9 вычислений функции, включая значение в конечной середине.

7 Числа Фибоначчи и выбор числа шагов

Введём нумерацию

$$F_0 = 0, \qquad F_1 = 1, \qquad F_j = F_{j-1} + F_{j-2} \quad (j \geq 2).$$

Начальные значения приведены в таблице.

Индекс j	0	1	2	3	4	5	6	7	8	9	10
F_j	0	1	1	2	3	5	8	13	21	34	55

Согласование с учебником. В [1, § 5.1.6] используется нумерация, начинающаяся с двух единиц при индексах 0 и 1. Наше F_j при $j \geq 1$ соответствует числу учебника с индексом $j - 1$. Формулы нельзя переносить между этими соглашениями без изменения индексов.

В методе Фибоначчи заранее выбирается целое $n \geq 4$. Начальные точки определяются формулами

$$y_0 = a_0 + \frac{F_{n-2}}{F_n} L_0, \qquad z_0 = a_0 + \frac{F_{n-1}}{F_n} L_0.$$

Сумма двух дробей равна единице, поэтому точки симметричны относительно середины. После сокращения число n в текущей схеме заменяется на $n - 1$. На следующих шагах пропорции меняются, но одно значение функции продолжает использоваться повторно.

Удобно обозначить оставшийся индекс через m. В начале $m = n$; на обычных шагах $m > 3$. Перед сравнением

$$y = a + \frac{F_{m-2}}{F_m}(b - a), \qquad z = a + \frac{F_{m-1}}{F_m}(b - a).$$

На каждом обычном шаге сохраняется доля F_{m-1}/F_m текущей длины. При последовательном перемножении промежуточные множители сокращаются:

$$L_k = \frac{F_{n-k}}{F_n} L_0, \qquad 0 \le k \le n - 3.$$

После $n - 3$ обычных сокращений $m = 3$ и длина равна $2L_0/F_n$. Следующий шаг требует особого правила, поскольку $F_1/F_3 = F_2/F_3 = 1/2$: две формальные пробные точки совпадают в середине.

8 Переходы метода Фибоначчи

Рассмотрим обычный шаг при $m > 3$. Если $f(y) \le f(z)$, сохраняется левая часть $[a; z]$. Её новая длина равна

$$L_{\text{new}} = \frac{F_{m-1}}{F_m} L.$$

Старая точка y становится новой правой точкой, поскольку

$$\frac{y - a}{L_{\text{new}}} = \frac{F_{m-2}}{F_{m-1}}.$$

Это как раз требуемое положение правой точки для нового индекса $m - 1$. Если следующий шаг ещё обычный, новая левая точка строится по формуле

$$y_{\text{new}} = a + \frac{F_{m-3}}{F_{m-1}} L_{\text{new}}.$$

Если $f(y) > f(z)$, сохраняется $[y; b]$, а старая точка z становится новой левой. Её положение в новом отрезке определяется разностью чисел Фибоначчи:

$$\frac{z - y}{L_{\text{new}}} = \frac{F_{m-1} - F_{m-2}}{F_{m-1}} = \frac{F_{m-3}}{F_{m-1}}.$$

Новая правая точка в этой ветви равна

$$z_{\text{new}} = a_{\text{new}} + \frac{F_{m-2}}{F_{m-1}} L_{\text{new}}.$$

После каждой ветви уменьшают m на единицу. Если получено $m = 3$, обычные формулы больше не применяются. Уже известная точка является серединой оставшегося отрезка; её значение нужно сохранить для заключительного сравнения.

Таким образом, рекуррентное равенство $F_j = F_{j-1} + F_{j-2}$ обеспечивает не только последовательность длин, но и правильное расположение повторно используемых точек. Именно это связывает числа Фибоначчи с экономией вычислений функции.

9 Заключительное смещение и оценка точности

Обозначим $h = L_0/F_n$. Перед последним сравнением отрезок имеет длину $2h$, а его середина c уже вычислена. Выберем малое положительное смещение $\eta < h$ и положим

$$y = c, \qquad z = c + \eta.$$

Если $f(c) \le f(c+\eta)$, сохраняем $[a; c+\eta]$ длиной $h+\eta$. Иначе сохраняем $[c; b]$ длиной h. В обоих случаях

$$L_{\text{final}} \le \frac{L_0}{F_n} + \eta.$$

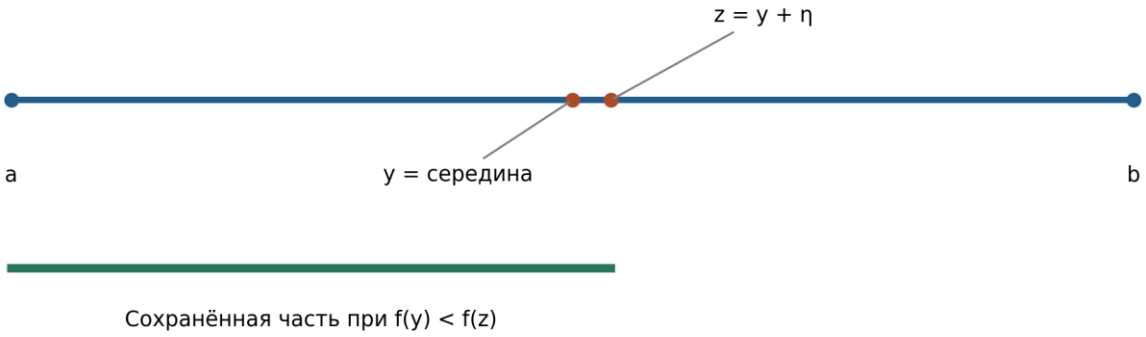


Рисунок 3 — Последнее смещение разделяет совпадающие пробные точки

Чтобы середина конечного отрезка имела погрешность не более ε_x, достаточно заранее обеспечить

$$\frac{L_0}{F_n} + \eta \le 2\varepsilon_x.$$

В учебной реализации принято $\eta = \varepsilon_x/10$. Выбирается минимальное $n \ge 4$, для которого

$$F_n \ge \frac{L_0}{2\varepsilon_x - \eta}.$$

Если исходный отрезок ещё требует сокращения, такой выбор обеспечивает $\eta < h$: при $n = 4$ имеем $h = L_0/3 > 2\varepsilon_x/3$; при $n > 4$ из минимальности индекса и $F_n < 2F_{n-1}$ следует $h > (2\varepsilon_x - \eta)/2$. Поэтому последняя новая точка остаётся внутри отрезка в точной арифметике.

Почему нельзя игнорировать смещение. Формула L_0/F_n описывает предельную длину при $\eta \to 0$. При положительном смещении левая ветвь может дать более длинный отрезок. Если выбирать n только по условию $L_0/F_n \le 2\varepsilon_x$, то учёт η способен нарушить требуемую точность. В программе проверяется и фактическая конечная длина.

Всего выполняется $n-3$ обычных сравнений и одно заключительное, то есть $k = n-2$ сокращений. Первое сравнение использует два значения, остальные — по одному новому. Поэтому

$$N_{\text{search}} = n - 1, \qquad N_f = n,$$

где второе равенство включает отдельный расчёт функции в конечной середине. Эти формулы предполагают штатное выполнение всей схемы без досрочной остановки из-за совпадения машинных значений или потери различимости точек.

10 Подробный пример метода Фибоначчи

Решим ту же задачу $f(x) = x^2 + 2/x$ на $[0{,}5; 3]$ при $\varepsilon_x = 0{,}05$. Унимодальность и точное решение уже установлены в разделе 6. Примем $\eta = 0{,}005$.

Планирование. Необходимо выбрать число Фибоначчи, удовлетворяющее

$$F_n \geq \frac{2{,}5}{0{,}1 - 0{,}005} = \frac{2{,}5}{0{,}095} \approx 26{,}315789.$$

Число $F_8 = 21$ недостаточно, а $F_9 = 34$ подходит. Поэтому $n = 9$, $h = 2{,}5/34 \approx 0{,}073529412$. План: 6 обычных сравнений, одно заключительное и 9 вычислений функции с учётом конечной середины.

Первое сравнение.

$$y_0 = 0{,}5 + \frac{13}{34} \cdot 2{,}5 \approx 1{,}455882,$$

$$z_0 = 0{,}5 + \frac{21}{34} \cdot 2{,}5 \approx 2{,}044118.$$

Получаем $f(y_0) \approx 3{,}493331 < f(z_0) \approx 5{,}156834$. Сохраняем $[0{,}5; 2{,}044118]$, старую левую точку переносим вправо, текущий индекс уменьшаем до $m = 8$.

Второе сравнение. Новая левая точка равна

$$y_1 = 0{,}5 + \frac{F_6}{F_8}(2{,}044117647 - 0{,}5) \approx 1{,}088235.$$

Её значение 3,022094 меньше сохранённого правого значения 3,493331. Получаем отрезок $[0{,}5; 1{,}455882]$. Продолжаем, уменьшая индекс m после каждого обычного сокращения.

Шаг	y	z	$f(y)$	$f(z)$	Новый отрезок
1	1,45588	2,04412	3,493331	5,156834	[0,5; 2,04412]
2	1,08824	1,45588	3,022094	3,493331	[0,5; 1,45588]
3	0,86765	1,08824	3,057896	3,022094	[0,86765; 1,45588]
4	1,08824	1,23529	3,022094	3,144999	[0,86765; 1,23529]
5	1,01471	1,08824	3,000643	3,022094	[0,86765; 1,08824]
6	0,94118	1,01471	3,010813	3,000643	[0,94118; 1,08824]
7	1,01471	1,01971	3,000643	3,00115	[0,94118; 1,01971]

После шестого сравнения остаётся

$$[a; b] \approx [0{,}941176471; 1{,}088235294], \qquad b - a = 2h.$$

Ранее вычисленная точка $c \approx 1{,}014705882$ является его серединой. Строим $z = c + 0{,}005 \approx 1{,}019705882$. Имеем $f(c) \approx 3{,}000642520 < f(z) \approx 3{,}001149957$. Поэтому конечный отрезок

$$[a_{\text{final}}; b_{\text{final}}] \approx [0{,}941176471; 1{,}019705882].$$

Его середина $\hat{x} \approx 0{,}980441176$, радиус $0{,}039264706 < 0{,}05$, значение функции $f(\hat{x}) \approx 3{,}001162906$. Фактическая ошибка координаты $0{,}019558824$ не превосходит радиуса. Оба метода выполнили по 9 вычислений функции, но дали разные конечные середины.

11 Сравнение методов при одинаковых условиях

Сравнивать методы по числу итераций недостаточно. Одна итерация дихотомии использует два новых значения, а золотое сечение и Фибоначчи после старта — одно. Основная сопоставимая характеристика здесь — число обращений к функции при одинаковом допуске по координате.

Метод	Поисковые вычисления	Длина после стандартных шагов
Дихотомия со смещением δ от середины	$2k$	$2\delta + (L_0 - 2\delta)/2^k$
Золотое сечение	$k + 1$ при $k \geq 1$	$\tau^k L_0$
Фибоначчи с последним смещением η	$n - 1$	Не более $L_0/F_n + \eta$

В таблице не учитывается дополнительное вычисление в конечной середине. Параметр δ дихотомии и последнее смещение η Фибоначчи имеют разные роли: первый влияет на каждое сокращение, второй используется только на заключительном шаге.

Для одного и того же числа поисковых вычислений N золотое сечение даёт относительную длину τ^{N-1}. В принятой нумерации метод Фибоначчи соответствует $n = N + 1$ и в пределе малого последнего смещения даёт $1/F_{N+1}$ [1].

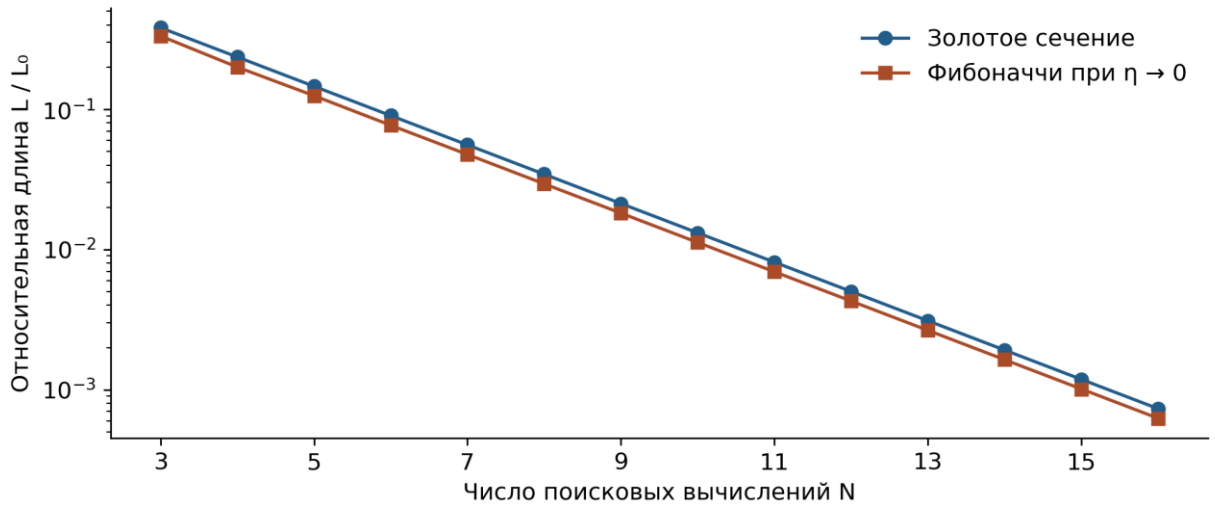


Рисунок 4 — Теоретическая относительная длина при одинаковом числе вычислений

График Фибоначчи показывает предел при $\eta \to 0$, а не точную длину программы с положительным смещением. При реальном расчёте используется оценка с добавкой η/L_0.

Золотое сечение удобно, когда поиск останавливают по фактической длине и не хотят заранее составлять конечную схему. Метод Фибоначчи удобен, когда заранее планируют число вычислений или подбирают его под заданный допуск. Отношение F_{m-1}/F_m стремится к τ при росте m, поэтому начальные пропорции длинной схемы Фибоначчи близки к золотому сечению.

В нашем примере радиус отрезка Фибоначчи меньше, но фактическая ошибка его середины больше. Противоречия нет: радиус является верхней оценкой возможной ошибки. Более узкий отрезок не гарантирует, что его середина в конкретном расчёте ближе к точному минимуму, чем середина другого отрезка.

12 Реализация и работа с программой

Архив к лекции содержит файл interval_methods.py с двумя реализациями, run_examples.py для запуска примера и индивидуальных вариантов, таблицу параметров variants.json и инструкцию README.txt. Требуется Python 3.10 или новее; дополнительные библиотеки для запуска этих файлов не нужны.

Сначала выполните вручную две итерации, затем запустите тот же пример. Сопоставляйте координаты пробных точек, знак сравнения и новые границы. Сравнение только конечного ответа не позволяет обнаружить повторные вычисления функции или ошибочный счётчик итераций.

12.1 Перенос значения функции

Ключевой фрагмент золотого сечения в левой ветви выглядит так:

```
b = z
z, fz = y, fy
if (b - a) / 2 <= eps_x:
    # Допуск достигнут; новое значение не нужно.
    break
y = a + (1 - tau) * (b - a)
fy = f(y)
```

Здесь правой точкой становится старая левая вместе с уже вычисленным значением. Важно соблюдать порядок присваиваний: если сначала перезаписать y, можно случайно потерять нужную координату.

Для правой ветви применяется симметричный переход:

```
a = y
y, fy = z, fz
if (b - a) / 2 <= eps_x:
    break
z = a + tau * (b - a)
fz = f(z)
```

Эти фрагменты показывают идею шага. Полная реализация в архиве дополнительно проверяет входные параметры, конечность значений, порядок пробных точек и ограничения числа итераций.

12.2 Подготовка схемы Фибоначчи

Для нетривиального исходного отрезка выбор индекса можно записать так:

```
eta = eps_x / 10
F = [0, 1, 1, 2, 3]
while (b - a) / F[-1] > 2 * eps_x - eta:
    F.append(F[-1] + F[-2])
n = len(F) - 1
```

Индекс n не равен количеству обычных сравнений. В принятой схеме обычных сравнений n - 3, всех сокращений n - 2, поисковых вычислений n - 1. Последний шаг со смещением реализован отдельно после обычного цикла.

При машинной арифметике ранее вычисленная центральная точка может отличаться на несколько единиц последнего разряда от заново рассчитанного (a + b) / 2. Программа сохраняет прежнюю точку и её значение, а после заключительного сравнения проверяет фактическую длину отрезка.

12.3 Запуск и чтение результата

Для запуска примера распакуйте архив в отдельную папку и выполните в её терминале:

```
python run_examples.py
```

В Windows вместо python можно использовать py. Для варианта 7 выполните два расчёта:

```
python run_examples.py --variant 7 --eps 0.001
python run_examples.py --variant 7 --eps 0.0001
```

Программа выводит приближение, значение функции, радиус конечного отрезка, фактическую ошибку относительно учебного точного ответа, число сокращений и счётчик обращений к функции. Таблицы шагов сохраняются в CSV, полный результат — в JSON. Повторный запуск с теми же параметрами перезаписывает файлы этого расчёта.

Основные поля результата: x — конечная середина; a, b — границы; radius — половина длины; iterations — число выполненных сокращений; n_f — все вычисления функции, включая конечную середину. У метода Фибоначчи также возвращаются n и eta.

Статус converged означает выполнение допуска по фактическому радиусу. Статус indistinguishable_values означает совпадение машинных значений в двух точках, precision_limit — невозможность продолжить схему с различимыми точками. У золотого сечения возможен max_iter; у Фибоначчи — roundoff_limit, если после всех запланированных шагов фактический радиус из-за округления оказался больше допуска. Описание всех статусов имеется в инструкции.

Программы не проверяют унимодальность автоматически и не используют интервальную арифметику. Теоретическое сохранение минимума предполагает достоверные сравнения функции. При слишком малом допуске следует исследовать точность вычисления самой функции, а не объявлять найденный результат точным по числу напечатанных знаков.

13 Самостоятельная работа

Для закрепления темы используются 18 вариантов с тремя семействами функций. Параметры s, p, c приведены в таблице; во всех вариантах $p > 0$.

Семейство A:

$$f(x) = (x - s)^2 + \frac{2p^3}{x - s} + c, \qquad x \in [s + p/2; s + 3p].$$

Семейство B:

$$f(x) = (x - s) - p\ln(x - s) + c, \qquad x \in [s + p/4; s + 4p].$$

Семейство C:

$$f(x) = e^{x-s} - p(x - s) + c, \qquad x \in [s - 2; s + 2].$$

№	Семейство	s	p	c
1	A	-2	0,5	0
2	A	-1	0,8	1
3	A	0	1,2	-1
4	A	1	1,5	2
5	A	2	2	-2
6	A	3	2,5	3
7	B	-2	0,5	0
8	B	-1	0,8	1
9	B	0	1,2	-1
10	B	1	1,5	2
11	B	2	2	-2
12	B	3	2,5	3
13	C	-2	0,5	0
14	C	-1	0,8	1
15	C	0	1,2	-1
16	C	1	1,5	2
17	C	2	2	-2
18	C	3	2,5	3

Порядок выполнения.

1. Запишите функцию и границы своего варианта. Проверьте область определения, найдите производную и обоснуйте единственность минимума на отрезке.
2. Для $\varepsilon_x = 10^{-3}$ определите число сокращений золотого сечения. Для Фибоначчи примите $\eta = \varepsilon_x/10$ и найдите минимальное допустимое $n \geq 4$.
3. Выполните вручную первые два сравнения каждого метода. В таблице укажите старые границы, две точки, значения функции, сохраняемый отрезок и число новых вычислений.
4. Запустите оба алгоритма и сравните начало программной таблицы с ручным решением. Проверьте содержание точного минимума во всех сохранённых отрезках.

5. Повторите расчёт с $\varepsilon_x = 10^{-4}$. Сопоставьте число сокращений, n_f, радиус и фактическую ошибку; объясните, почему расходы увеличились.
6. Укажите, включено ли вычисление в конечной середине в ваш счётчик. Проверьте теоретические формулы для количества обращений.
7. Сделайте вывод о точности и вычислительной стоимости методов для своего варианта. Не объявляйте метод лучшим по единственному совпадению последних знаков ответа.

Содержание краткого отчёта: постановка задачи; обоснование унимодальности; расчёт параметров; ручные таблицы; программные результаты при двух допусках; проверка оценки ошибки; вывод. Листинг всей программы в отчёт вставлять необязательно: достаточно собственных изменений и ключевого фрагмента переноса точки.

13.1 Ответы для самопроверки

Для семейств A и B точная точка минимума равна

$$x^* = s + p.$$

Для семейства A значение в минимуме $f(x^*) = 3p^2 + c$, а для B — $f(x^*) = p - p\ln p + c$. Для семейства C

$$x^* = s + \ln p, \qquad f(x^*) = p - p\ln p + c.$$

Обоснование получается заменой $t = x - s$. Для A имеем $f'(x) = 2t - 2p^3/t^2$ и $f''(x) = 2 + 4p^3/t^3 > 0$ при $t > 0$. Для B производная $1 - p/t$ обращается в нуль при $t = p$, а вторая производная p/t^2 положительна. Для C производная $e^t - p$ обращается в нуль при $t = \ln p$, а вторая производная e^t положительна.

У всех заданных вариантов эти точки лежат внутри указанных отрезков. В семействе C параметры выбраны так, что $-2 < \ln p < 2$. Следовательно, сравнение с аналитическим ответом корректно для каждого варианта.

14 Контрольные вопросы

1. Какое свойство функции позволяет исключать часть отрезка по двум значениям?
2. Почему точное равенство значений и совпадение чисел float требуют разного толкования?
3. Как из требования повторного использования точки получается уравнение $\tau^2 + \tau = 1$?
4. Какие координаты и значения переносятся в каждой ветви золотого сечения?
5. Почему при допуске по координате ε_x допускается конечная длина $2\varepsilon_x$?
6. Как вычисляется число сокращений золотого сечения и почему меняется знак неравенства при логарифмировании?
7. Чем число сравнений отличается от числа вычислений функции?
8. Какое соглашение об индексах чисел Фибоначчи используется в лекции?
9. Почему длины отрезков в методе Фибоначчи выражаются отношениями соседних чисел?
10. На каком этапе пробные точки совпадают и как выполняется последнее сравнение?
11. Почему в оценке конечной длины необходимо учитывать η?
12. Может ли меньший радиус сопровождаться большей фактической ошибкой середины? Объясните на разобранном примере.

Задача для обсуждения. Пусть для золотого сечения $L_0 = 4$ и $\varepsilon_x = 0{,}01$. Найдите число сокращений и полное число вычислений функции при отдельном расчёте в конечной середине. Затем выберите индекс Фибоначчи при $\eta = 0{,}001$.

Ответ: золотое сечение требует 12 сокращений и 14 вычислений. Для Фибоначчи необходимо $F_n \geq 4/0{,}019 \approx 210{,}5263$; подходит $F_{13} = 233$. Получаем 11 сокращений и 13 вычислений с учётом конечной середины. Гарантированный радиус не превосходит $(4/233 + 0{,}001)/2 \approx 0{,}00908369$.

15 Литература

[1] Пантелеев А. В., Летова Т. А. Методы оптимизации в примерах и задачах: учебное пособие. 4-е изд., испр. СПб.: Лань, 2015. 512 с. Для изучения темы: § 5.1.5 «Метод золотого сечения», с. 106–110; § 5.1.6 «Метод Фибоначчи», с. 110–113. В лекции используется нумерация чисел Фибоначчи с $F_0 = 0$, $F_1 = 1$ и явно учитывается заключительное смещение.

[2] Рабочая программа дисциплины «Методы оптимизации», 2024. Образовательная программа 6B06103 «Математическое и компьютерное моделирование». Тема 8: метод золотого сечения и метод Фибоначчи.