****

**Министерство науки и высшего образования Российской Федерации**

**МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ им. Н. Э. БАУМАНА**

Факультет: «Информатика и системы управления»

Кафедра: «Информационная безопасность (ИУ8)» / ГУИМЦ

**ТЕОРИЯ СИСТЕМ И СИСТЕМНЫЙ АНАЛИЗ**

**Лабораторная работа №1 на тему:**

«Исследование методов прямого поиска экстремума унимодальной функции одного переменного»

Вариант 4

**Преподаватель:** Коннова Н. С.

**Студент:** Месяцева Н. В.

**Группа**: ИУ8Ц-51

Москва

2019

# **Цель работы**

Исследовать функционирование и провести сравнительный анализ различных алгоритмов прямого поиска экстремума (пассивный поиск, метод дихотомии, золотого сечения, Фибоначчи) на примере унимодальной функции одного переменного.

# **Постановка задачи**

На интервале задана унимодальная функция одного переменного . Используя *метод последовательного поиска дихотомию* найти интервал нахождения минимума при заданной наибольшей допустимой длине интервала неопределенности . Провести сравнение с *методом оптимального пассивного поиска*. Результат, в зависимости от числа точек разбиения N, представить в виде таблицы.

# **Ход работы**

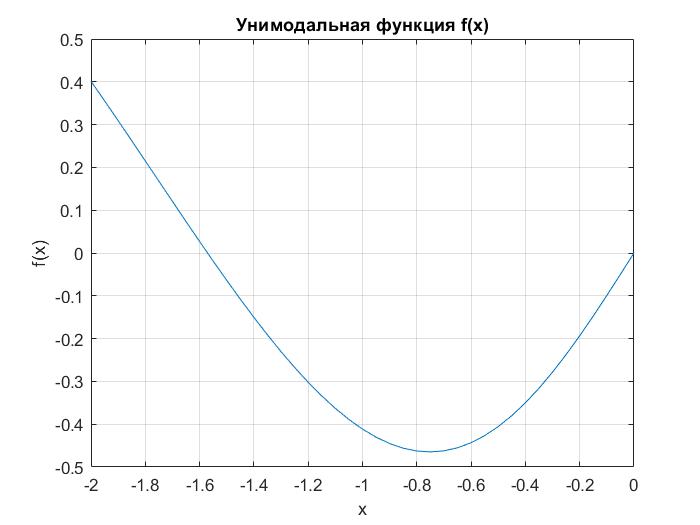
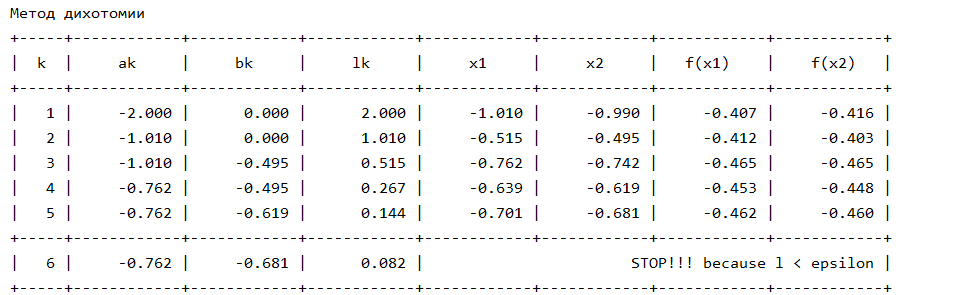
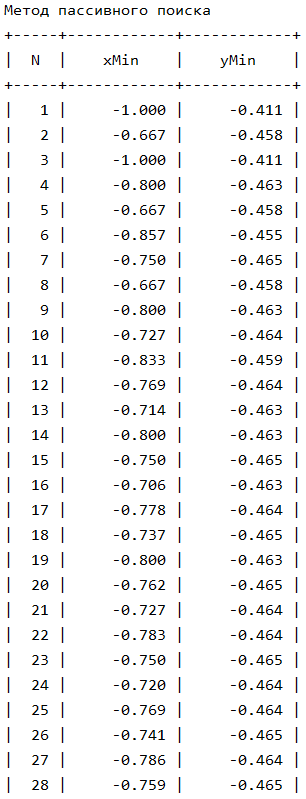
**Известно, что на интервале дана унимодальная функция (см. рис. 1). Сначала используем метод дихотомии, чтобы найти интервал нахождения минимума при заданной наибольшей допустимой длине интервала неопределённости

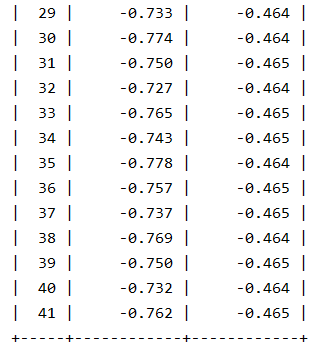
Рисунок 1. График унимодальной функции f(x) = cos(x)th(x)

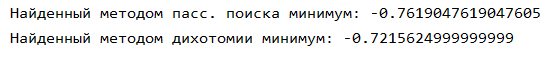
Рассмотрим последовательный поиск методом дихотомии. Примем очередной шаг (итерацию) за Тогда на отрезке ] длиной выберем две точки , где . Поскольку в нашем случае В этом методе на каждой итерации минимизируемая функция вычисляется дважды, а интервал неопределённости сокращается в два раза (при достаточно малых .

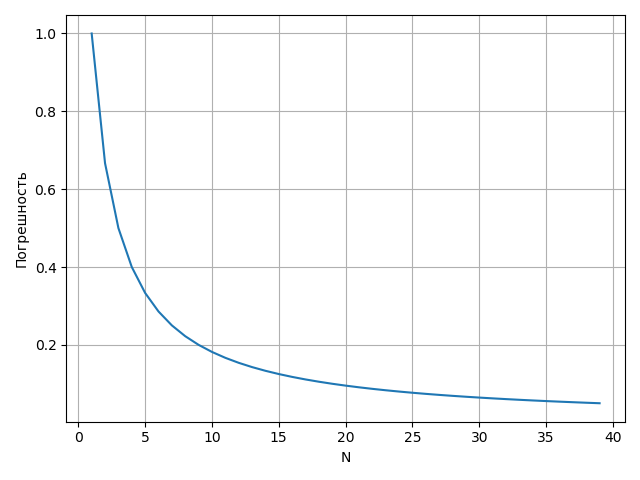
Результат программы приведён ниже:



Алгоритм оптимального пассивного поиска заключается в следующем: заданный отрезок поиска делится на N + 1 частей точками частей точками с координатами и вычисляется значение функции в каждой из этих точек, среди которых ищется экстремум. Точность поиска равна, наименьшая длина неопределённости .   
Результат программы:





Также приводится график зависимостей N от погрешности.  


## **Вывод**

Оба алгоритма реализуют поиск экстремума унимодальной функции одного переменного, но при сравнении их эффективности, выясняется, что метод дихотомии выгоднее использовании для отыскания минимума. Потому что в методе оптимального поиска все точки выбираются заранее, а в методе дихотомии для вычисления очередной точки используется информация, полученная на более ранней стадии расчетов. То есть в методе дихотомии не требуется вычислять значения функции в точках, принадлежащих отбрасываемым на каждом шаге полуинтервалам.

# **Листинг (Python)**

**from** math **import** cos, tanh  
**import** matplotlib.pyplot **as** plt *# Субмодуль для создания графиков***from** numpy **import** arange *# Импортируем субмодуль создания массива  
  
# Реализация метода пассивного поиска***def** PassiveSearch(f, a, b, Nmax):  
  
 print(**"Метод пассивного поиска"**)  
 print(**"+-----+------------+------------+"**)  
 print(**"| N | xMin | yMin |"**)  
 print(**"+-----+------------+------------+"**)  
  
 N = 1 *# Стартовая точка* inaccuracy = [] *# Пока что пустой список неточностей* N\_list = [] *# Пока что пустой список точек N  
  
# Цикл для нахождения минимума* **while** N <= Nmax:  
 accuracy = (b - a) / (N + 1) *# Шаг поиска* inaccuracy.append(accuracy) *# Добавляем неточность* N\_list.append(N) *# Добавляем N* xMin = a *# Считаем, что минимум находится в точке a* x = a  
 *# Идём по всему интервалу* **while** x <= b:  
 **if** f(x) < f(xMin): *# Если очередное значение стало меньше,* xMin = x *# то обновляем точку минимума* x += accuracy *# Переходим на следующую точку* print(**'| %\*d | %\*.3f | %\*.3f |'** % (3, N, 10, xMin, 10, f(xMin)))  
 N += 1 *# Увеличиваем итерацию на 1  
 # N \*= 2 # Для ускорения нахождения минимума, т.е. каждый раз умножая на 2* print(**"+-----+------------+------------+"**)  
  
  
*# Рисуем график зависимостей неточности от числа точек N* plt.plot(N\_list, inaccuracy, **'ro'**) *# Формируем ТОЧЕЧНЫЙ график по осям* plt.ylabel(**'Погрешность'**) *# Подписываем ось У* plt.xlabel(**'N'**) *# Подписываем ось Х* plt.grid(**True**) *# Включаем сетку* plt.show() *# Оторбажаем график  
  
 #return min* **return** xMin *# Возвращщаем конечную точку минимума  
  
# Реализация метода дихотомии***def** Dichotomy(f, a, b, eps):  
 delta = 0.01  
 k = 1  
 x = 0  
  
 print(**"Метод дихотомии"**)  
 print(**"+-----+------------+------------+------------+------------+------------+------------+------------+"**)  
 print(**"| k | ak | bk | lk | x1 | x2 | f(x1) | f(x2) |"**)  
 print(**"+-----+------------+------------+------------+------------+------------+------------+------------+"**)  
  
 *# Запускаем цикл. Пока интервал не сомкнётся...* **while** abs(b - a) >= eps:  
 x = (a + b) / 2  
  
 x\_Left = x - delta  
 x\_Right = x + delta  
  
 func\_xLeft = f(x\_Left)  
 func\_xRight = f(x\_Right)  
  
 print(**'| %\*d | %\*.3f | %\*.3f | %\*.3f | %\*.3f | %\*.3f | %\*.3f | %\*.3f |'** % (3, k, 10, a, 10, b, 10, b - a, 10, x\_Left, 10, x\_Right, 10, func\_xLeft, 10, func\_xRight))  
 *# Выполняем процедуру исключения отрезка* **if** func\_xLeft > func\_xRight:  
 a = x\_Left  
 **else**:  
 b = x\_Right  
  
 k += 1  
  
 print(**"+-----+------------+------------+------------+------------+------------+------------+------------+"**)  
  
 **if** abs(b - a) <= eps:  
 print(**'| %\*d | %\*.3f | %\*.3f | %\*.3f | %49s |'** % (  
 3, k, 10, a, 10, b, 10, b - a, **'STOP!!! because l < epsilon'** ))  
  
 print(**"+-----+------------+------------+------------+------------+------------+------------+------------+"**)  
  
 **return** (a + b) / 2 *# Возвращает вычисленный минимум  
  
# Заданная унимодальная функция для поиска минимума***if** \_\_name\_\_ == **"\_\_main\_\_"**:  
 UnimodalFunction = **lambda** x: cos(x) \* tanh(x)  
  
*# Интервал поиска [-2, 0]* a = -2  
 b = 0  
 eps = 0.1 *# Точность* Nmax = (2\*(b-a)/eps + 1) *# Максимально возможное число точек* delta = 0.01  
 xlist = arange(a, b, delta)  
 ylistUni = [UnimodalFunction(x) **for** x **in** xlist]  
  
 plt.plot(xlist, ylistUni)  
 plt.title(**"Унимодальная функция"**)  
 plt.grid(**True**)  
 plt.show()  
  
 xMin\_ps = PassiveSearch(UnimodalFunction, a, b, Nmax)  
 xMin\_dic = Dichotomy(UnimodalFunction, a, b, eps)  
  
print(**'Найденный методом пасс. поиска минимум:'**, xMin\_ps)  
print(**'Найденный методом дихотомии минимум:'**, xMin\_dic)