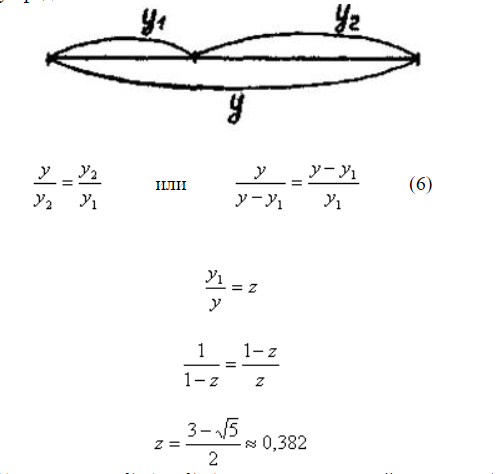
**Тема 2. Методы одномерной оптимизации.**

**Задание 1.** Написать (добавить в собственный класс/библиотеку) следующие функции:

1. Поиск экстремума функции одной переменной методом **золотого сечения**;
   1. на отрезке [a;b] мы берём точку x1,

так, чтобы он делил этот орезок на y1

и y2, где y/y1 = y1/y2= 0,618

* 1. Симметрично точке x1 отмечаем x2

Или применяем пункт a считая с

Изображение выглядит как текст, небо, антенна, провод

Автоматически созданное описаниедругой стороны

* 1. Далее вычисляем значение функции в точках x1 и x2
  2. Если f(x1)<f(x2)->Минимум реализуется на отрезке [a, x2]->

Запоминаем x2 вместо b

(f(x1)>f(x2) ->Минимум реализуется на отрезке [x1, b]-> запоминаем x1 вместо a)

* 1. Повторяем пока не достигнем требуемой точности

1. Поиск экстремума функции одной переменной методом **парабол**;
   1. Введите начальный шаг поиска h, задайте начальное значение X0, погрешность результата e.
   2. Вычислить f(x0) и f(x0+h).
   3. Если f(x0+h) > f(x0), то взять в качестве третьей точки X0-h и вычислить f(X0-h) [ x1=x0-h, x2=x0, x3=x0+h ].

В противном случае в качестве третьей точки взять X0+2h и найти f(X0+2h) [x1=x0, x2=x0+h, x3=x0+2h ],

* 1. Вычисляем y1=f(x1), y2=f(x2), y3=f(x3); Вычисляем a, b

a:=((y1-y3)\*z2-(y2-y3)\*z1)/(z1\*z2\*(z1-z2)); b:=((y1-y3)\*z2\*z2-(y2-y3)\*z1\*z1)/(z1\*z2\*(z2-z1)),

где z1:=x1-x3;z2:=x2-x3;c:=y3

* 1. Проверяем a>0, если нет, то начальное приближение выбрано неудачно и следует закончить вычисление с таким значением и сделать два шага в сторону убывания функции x0=x0+2h и повторить снова с п. b, пока не выполнится a>0
  2. Вычисляем Zm (Zm= — b / (2\*a))
  3. Проверяем |Zm| < e, если да, то Xm=x3+Zm, Ym=f(Xm), конец.
  4. Если нет, переименовываем точки, отбрасывая точку х1: x1=x2, x2=x3, x3=x3+Zm, находим y1=y2, y2=y3, y3=f(x3).
  5. Переходим на п. e.

1. Поиск экстремума функции одной переменной комбинированным методом **Брента**;

Вход: Интервал оптимизации (a, c), точность ε;

Выход: Точка и значение минимума xmin, fmin;

Инициализация K = (3 −

√

5)/2, x = w = v = a + K(c − a), fx = fw = fv = f(x);

Инициализация длины текущего и предыдущего шага d = e = c − a;

пока Итерации до сходимости

Изображение выглядит как текст

Автоматически созданное описаниеg = e, e = d;

tol = ε|x| + ε/10 ;

если Выполнен критерий останова:



Выход из цикла;

если Точки x, w, v и значения fx, fw, fv – разные то

Параболическая аппроксимация, находим u – минимум параболы;

если u ∈ [a, c] и |u − x| < g/2 то

Принимаем u;

если u − a < 2tol или c − u < 2tol то

u = x − sign(x − (a + c)/2)tol;

если Парабола не принята то

если x < (a + c)/2 то

u = x + K(c − x); // Золотое сечение [x, c];

e = c − x;

иначе

u = x − K(x − a); // Золотое сечение [a, x];

e = x − a;

если |u − x| < tol то

u = x + sign(u − x)tol; // Задаём минимальную близость между u и x

d = |u − x|;

Вычисляем fu = f(u);

если fu ≤ fx то

если u ≥ x то

a = x;

иначе

c = x;

v = w, w = x, x = u, fv = fw, fw = fx, fx = fu;

иначе

если u ≥ x то

c = u;

иначе

a = u;

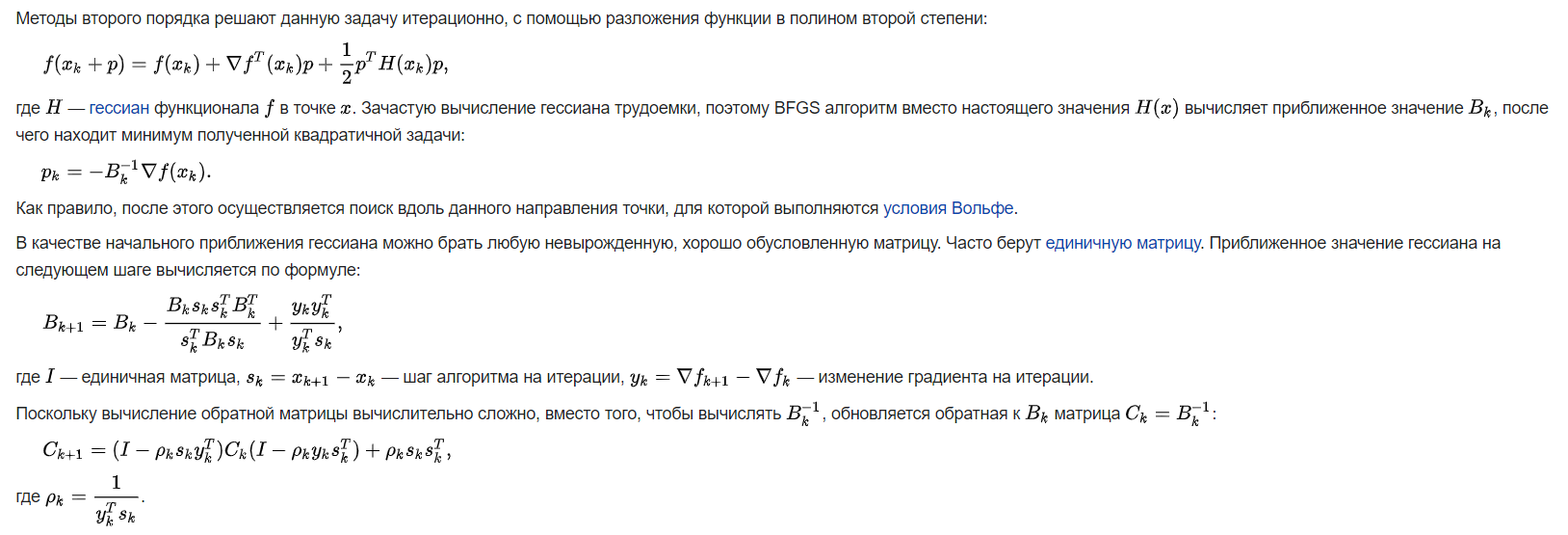
если fu ≤ fw или w = x то

v = w, w = u, fv = fw, fw = fu;

иначе если fu ≤ fv или v = x или v = w то

v = u, fv = fu;

1. Алгоритм неточной одномерной минимизации (**Алгоритм Бройдена — Флетчера — Гольдфарба — Шанно**);

Изображение выглядит как текст

Автоматически созданное описание

