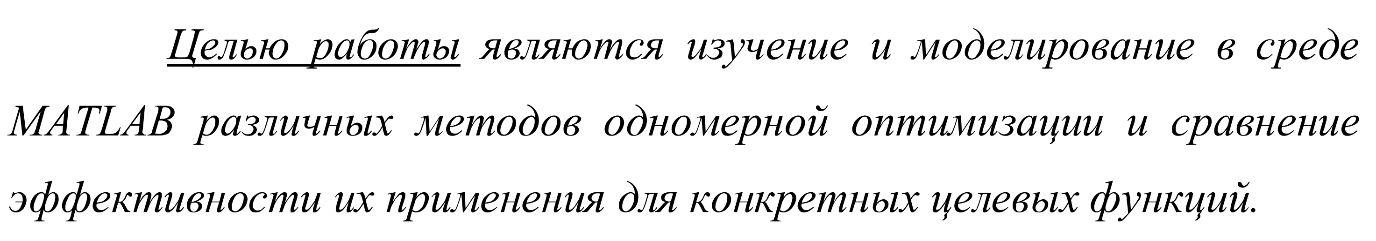
**Лабораторная работа № 1.**

Вариант № 8.

Отчет выполнил: Рыжов Ф., Елизавета И., Морозов И., Мясоедова А.



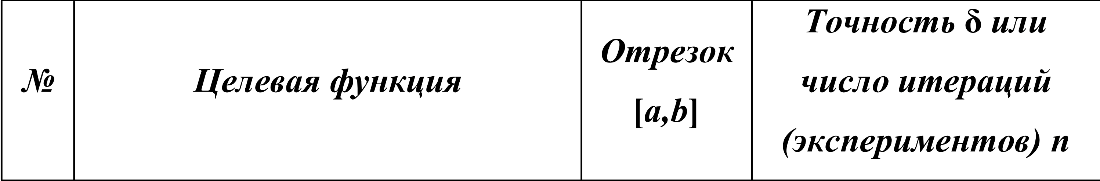
Всего будет рассмотрено 7 методов:

1. Метод перебора на равномерной сетке;
2. Метод деления отрезка по полам (метод дихотомии);
3. Метод золотого сечения;
4. Метод Фибоначчи;
5. Метод средней точки (метод Больцмана);
6. Метод касательных;
7. Метод парабол.

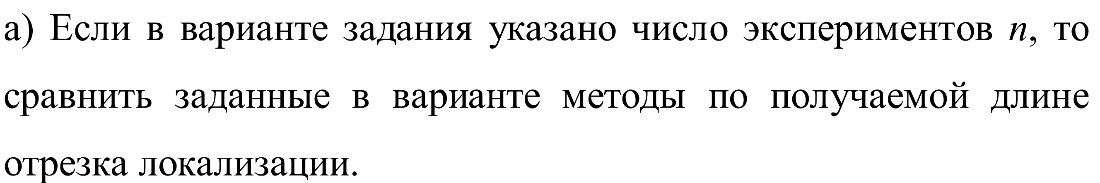
Методы 6-7 основаны на аппроксимации. Метод 5 основан на использовании производных.

Все рассмотренные номера реализовано на Mathlab 2022.

Начальные условия:

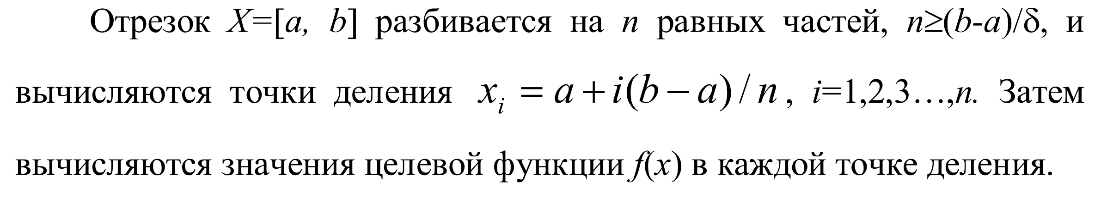




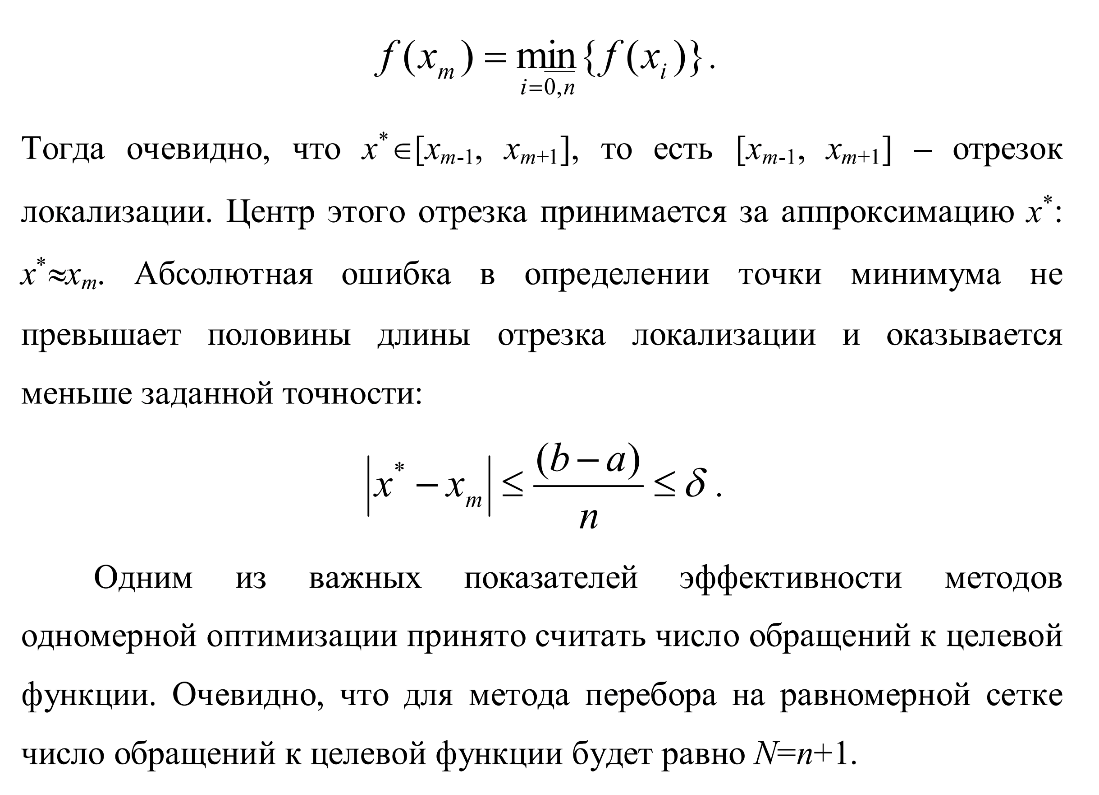


Метод перебора на равномерной сетке.

Алгоритм:







Код программы:

clear all

a = -1;

b = 0;

n = 18;

for i = 1: n

x(i) = a + i \* (b - a) / n;

end

f = x .^ (2) + exp(x);

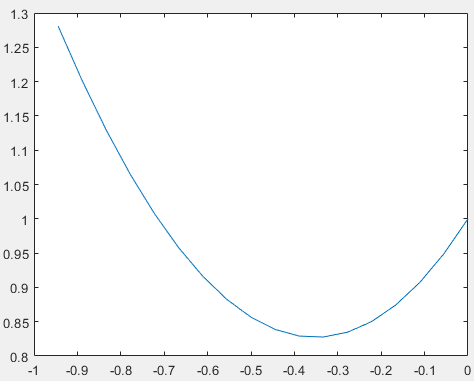
plot(x, f)

[i, j] = min(f);

fm = i

xm = x(j)

График, точка минимума и значение функции в ней:

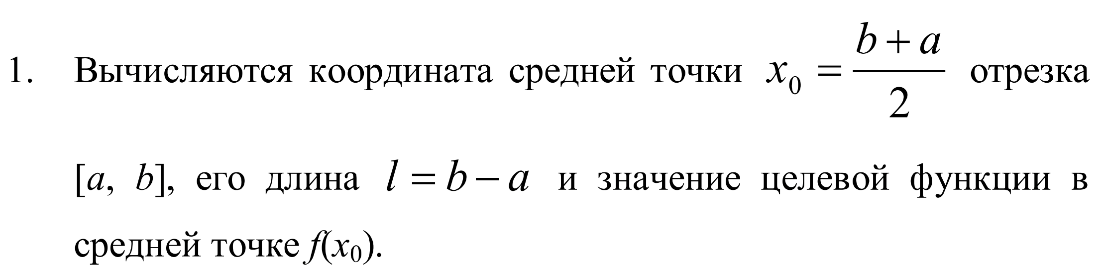


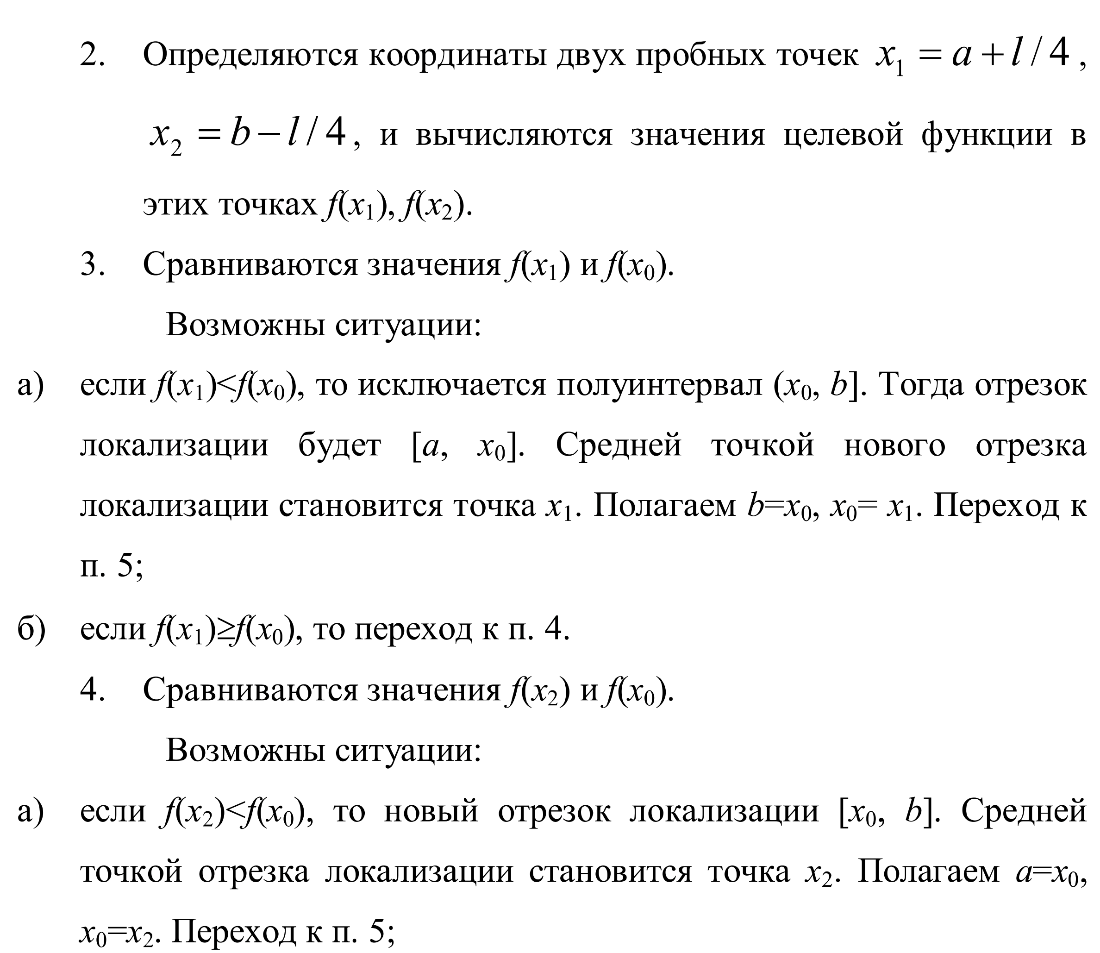


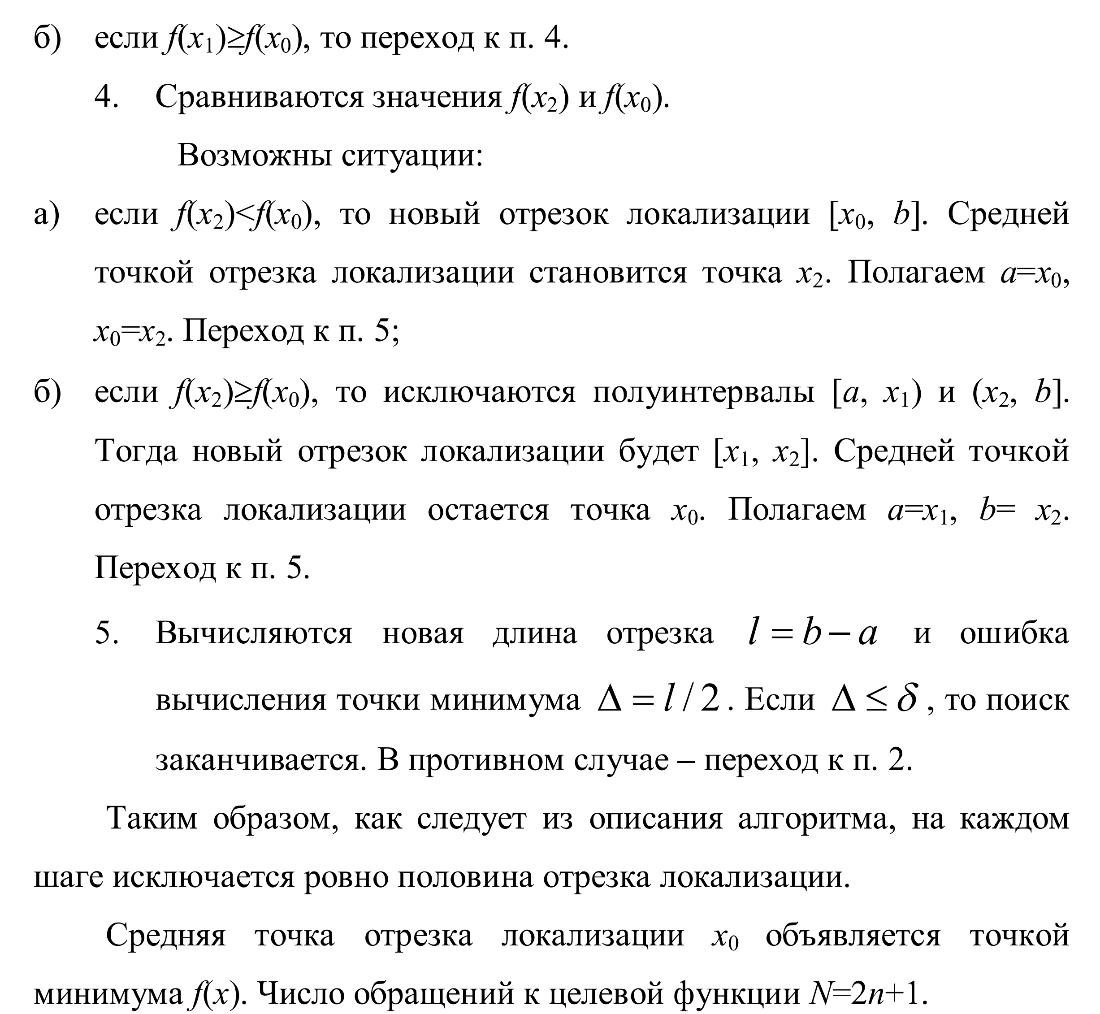


Метод деления отрезка по полам.

Алгоритм:







Код программы:

Файл function.mlx (наша функция по условиям):

function y = fun(x)

y = x .^2 + exp(x);

Файл solution.mlx (сборка всех файлов и запуск программы):

clear, clc

a = -1;

b = 0;

n = 18;

x0 = dixot('fun', a, b, n);

fm = fun(x0);

xm = x0;

fplot(@fun, [a, b]);

grid on

hold on

plot([a, b], fun(xm))

Файл dixot.mlx (фактически решение):

function x0 = dixot(fun, a, b, n)

for k = 1: n

x0 = (b + a) / 2;

l = b - a;

x1 = a + l/4;

x2 = b - l/4;

y = feval(fun, x0);

w = feval(fun, x1);

z = feval(fun, x2);

if w < y

b = x0 , x0 = x1

else

if z < y

a = x0, x0 = x2

else

a = x1, b = x2;

end;

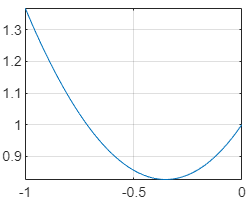
end;

l = abs(b - a);

x0 = (b + a) / 2;

end;

График, точка минимума и значение функции в ней:

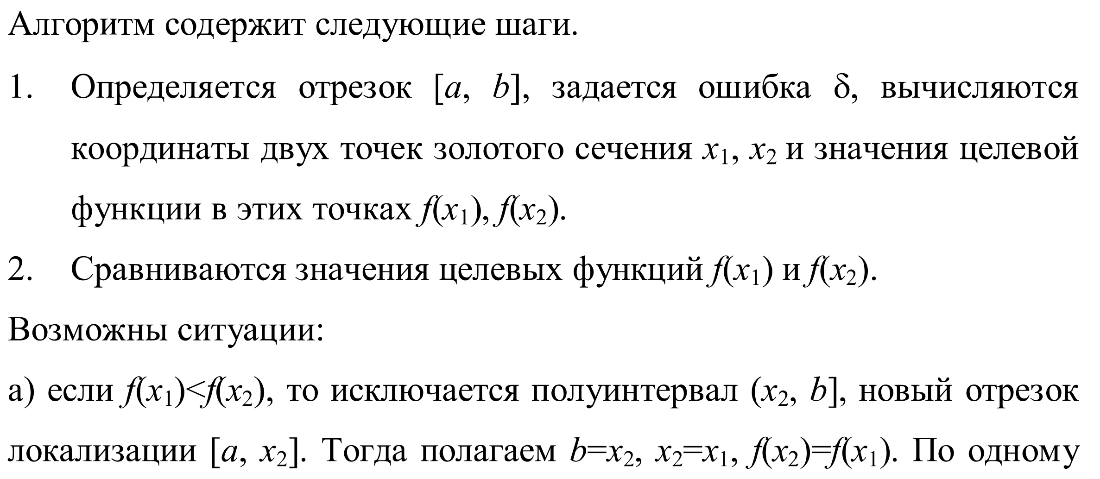


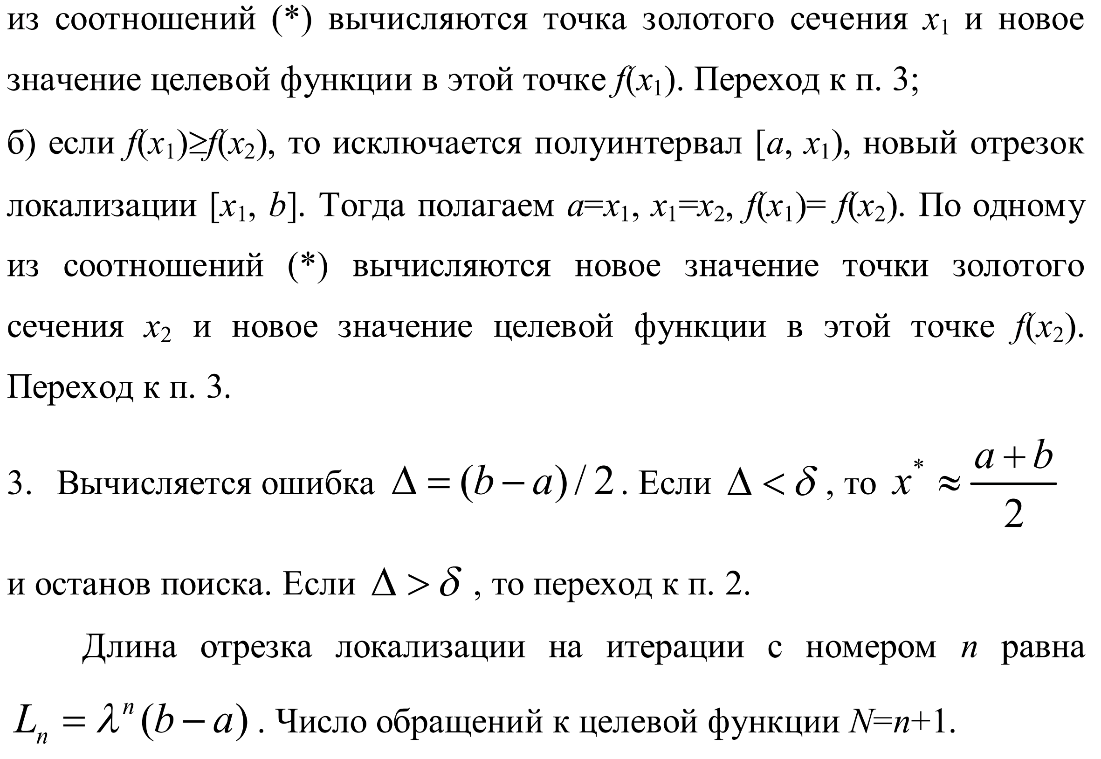




Метод золотого сечения.

Алгоритм:





Код программы:

Файл function.mlx (наша функция по условиям):

function y = fun(x)

y = x .^2 + exp(x);

Файл solution.mlx (сборка всех файлов и запуск программы):

clear, clc

a = -1;

b = 0;

n = 18;

x0 = gold('fun', a, b, n);

fplot(@fun, [a, b]);

grid on

hold on

plot([a, b], fun(x0))

fm = fun(x0);

xm = x0;

Файл gold.mlx (фактически решение):

function x0 = gold(fun, a, b, n)

L = (sqrt(5) - 1) / 2;

for k=1:n

x1 = a + (1 - L) \* (b - a);

x2 = a + L \* (b - a);

y = feval(fun, x1);

w = feval(fun, x2);

if y < w

b = x2, x2 = x1, w = y, x1 = a + b - x2, y = feval(fun, x1);

else

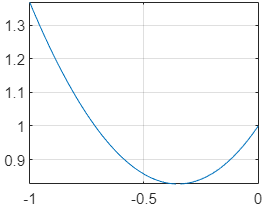
a = x1, x1 = x2, y = w, x2 = a + b - x1, w = feval(fun, x2);

end

x0 = (a + b) / 2, break;

end

График, точка минимума и значение функции в ней:

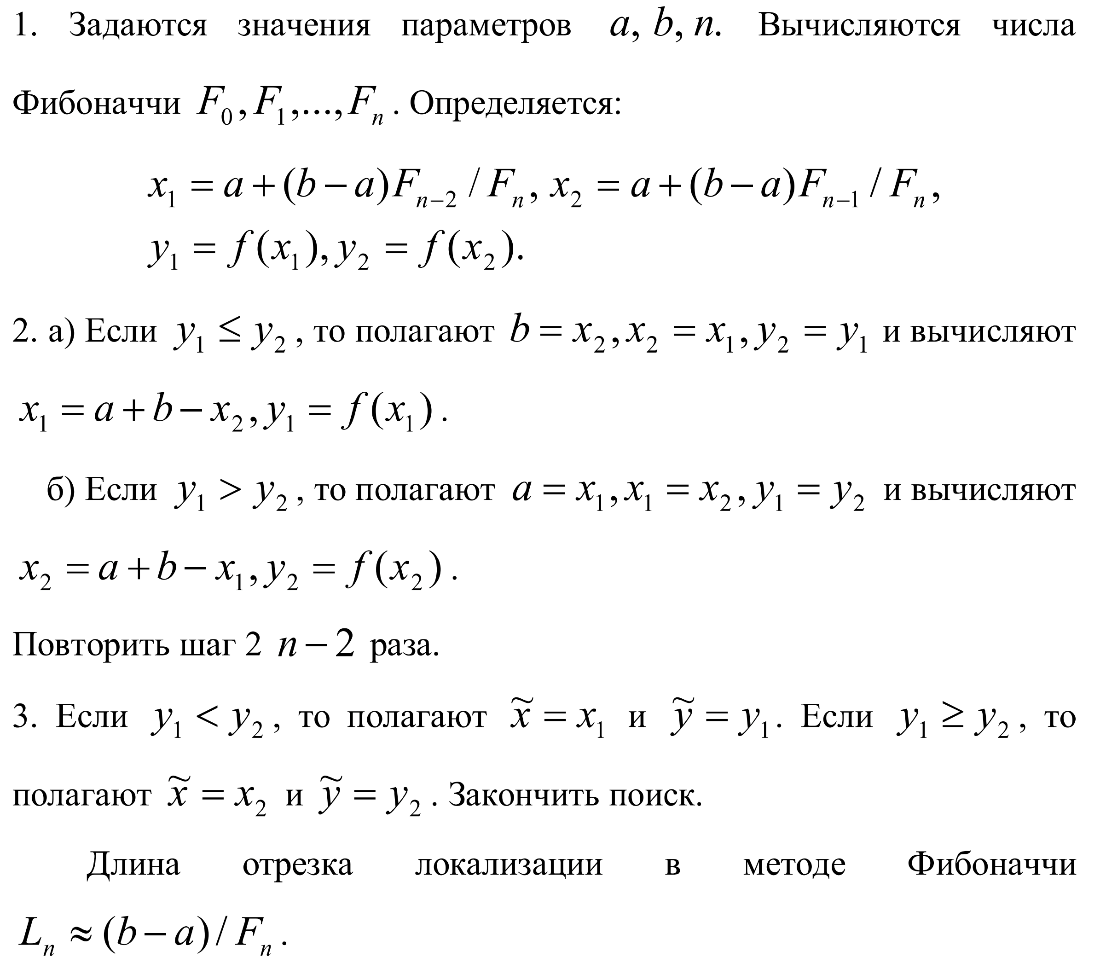






Метод Фибоначчи.

Алгоритм:



Код программы:

Файл function.mlx (наша функция по условиям):

function y = fun(x)

y = x .^2 + exp(x);

Файл solution.mlx (сборка всех файлов и запуск программы):

clear, clc

a = -1;

b = 0;

n = 18;

fib = [0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765, 10946, 17711]

x0 = fbi('fun', a, b, n, fib);

fplot(@fun, [a, b]);

grid on

hold on

plot([a, b], fun(x0))

fm = fun(x0);

xm = x0;

Файл fbi.mlx (фактически решение):

function x0 = fbi(fun, a, b, n, fib)

for k = 1: n - 2

x1 = a + (b - a) \* fib(n-2) / fib(n)

x2 = a + (b - a) \* fib(n-1) / fib(n)

y = feval(fun, x1);

z = feval(fun, x2);

if y <= z

b = x2, x2 = x1, z = y, x1 = a + b - z, y = feval(fun, x1);

else

a = x1, x1 = x2, y = z, x2 = a + b - x1, z = feval(fun, x2);

end

if y > z

x0 = x1

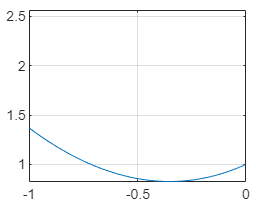
else

x0 = x2

end

end

График, точка минимума и значение функции в ней:

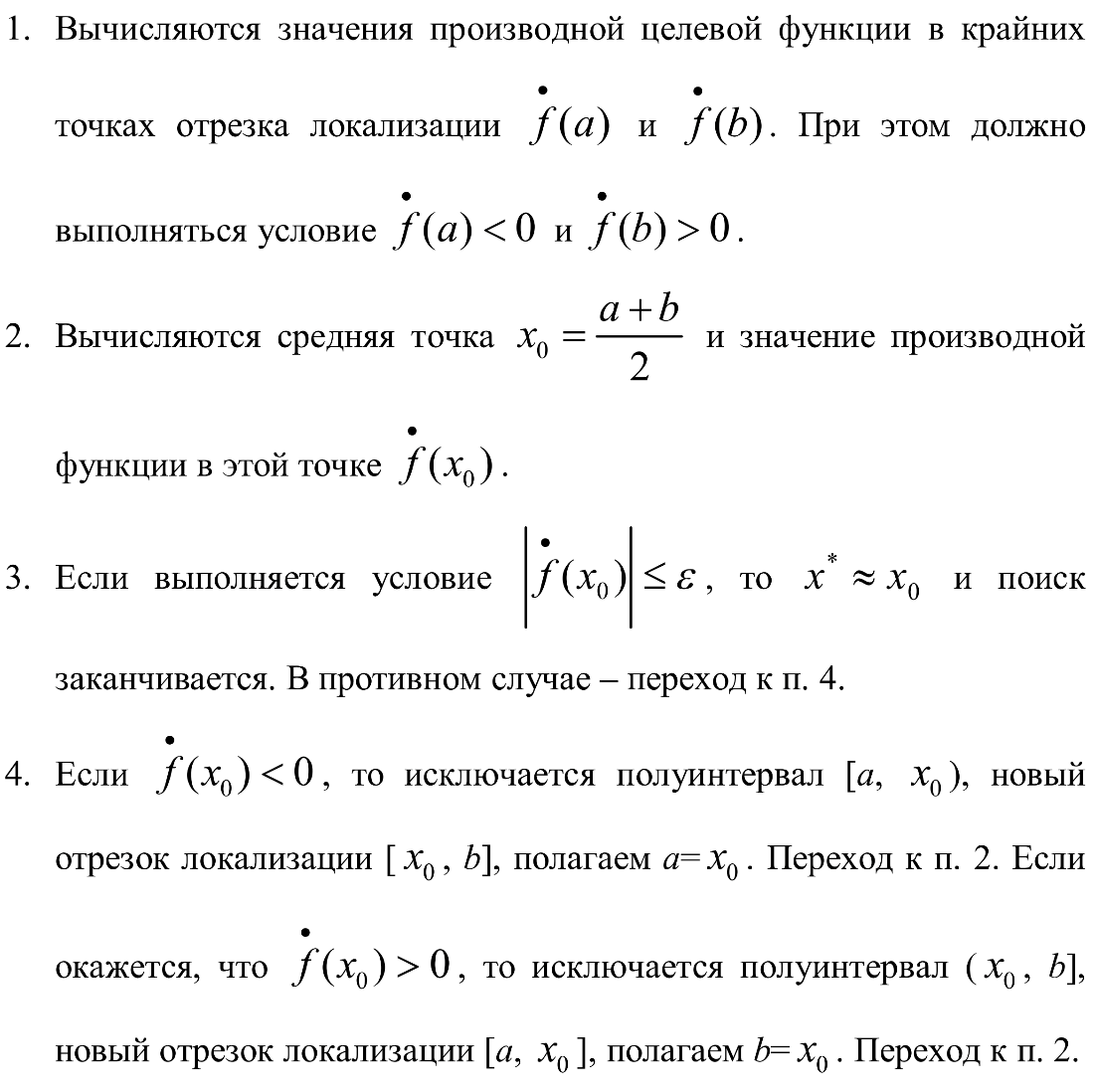






Метод средней точки.

Алгоритм:



Код программы:

Файл function.mlx (наша функция по условиям):

function y = fun(x)

y = x .^2 + exp(x);

Файл solution.mlx (сборка всех файлов и запуск программы):

clear, clc

a = -1;

b = 0;

n = 18;

x0 = opt('fun', a, b, n)

fplot(@fun, [a, b]);

grid on

hold on

plot([a, b], fun(x0))

fm = fun(x0);

xm = x0;

Файл opt.mlx (фактически решение):

function x0 = opt(fun, a, b, n)

for k=1: n

y = 2 \* a - exp(-0.25 \* a)

z = 2 \* b - exp(-0.25 \* b)

x0 = (a + b) / 2

z = 2 \* x0 - exp(-0.25 \* x0)

if z < 0

a = x0

else

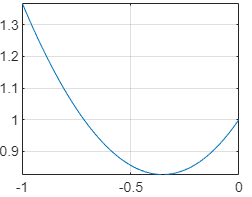
b = x0

end

end

end

График, точка минимума и знaчение функции в ней:

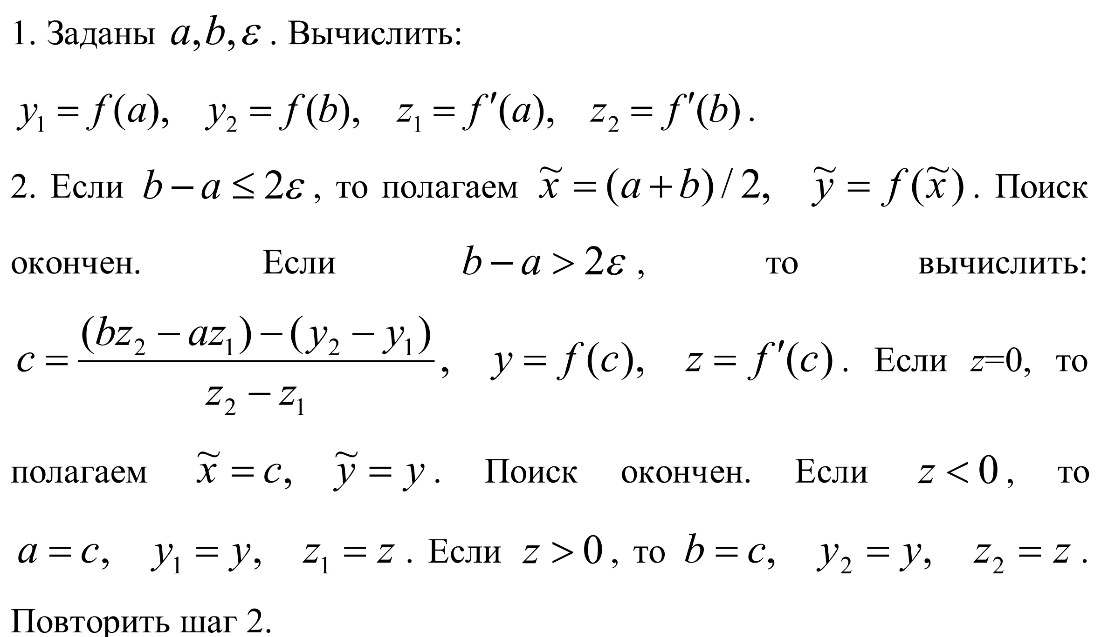






Метод касательных.

Алгоритм:



Код программы:

Файл function.mlx (наша функция по условиям):

function y = fun(x)

y = x .^2 + exp(x);

Файл solution.mlx (сборка всех файлов и запуск программы):

clear, clc

a = -1;

b = 0;

n = 18;

x0 = func('fun', a, b, n);

fplot(@fun, [a, b]);

grid on

hold on

plot([a, b], fun(x0))

fm = fun(x0);

xm = x0;

Файл func.mlx (фактически решение):

function x0 = func(fun, a, b, n)

for k =1: n

y1 = feval(fun, a);

y2 = feval(fun, b);

z1 = 2 \* a - exp(-0.25 \* a);

z2 = 2 \* b - exp(-0.25 \* b);

c = ((b \* z2 - a \* z1) - (y2 - y1)) / (z2 - z1), y = feval(fun, c), z = exp(c) - 10 \* sin(c)

x0 = (a + b) / 2;

if z == 0

x0 = c,

break;

if z < 0

a = c, y1 = y, z1 = z,

x0 = (a + b) / 2

else

b = c, y2 = y, z2 = z,

x0 = (a + b) / 2

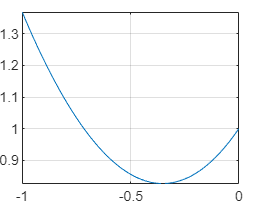
end

end

end

end

График, точка минимума и знaчение функции в ней:

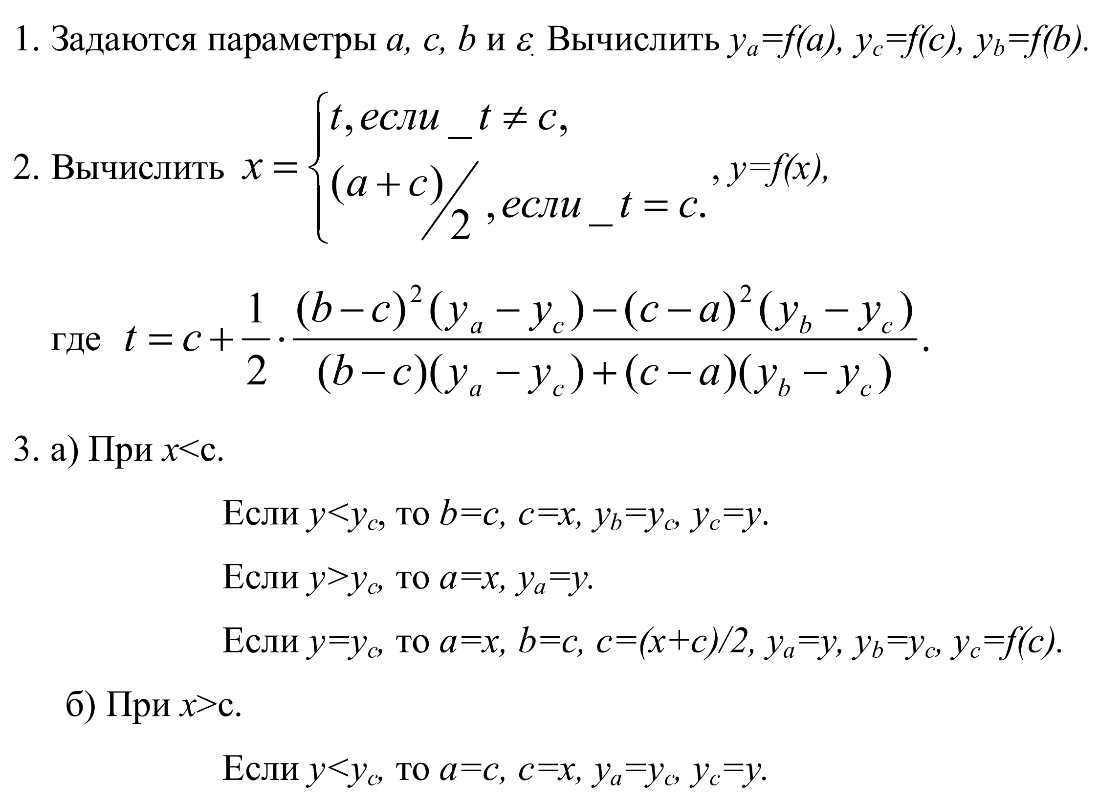


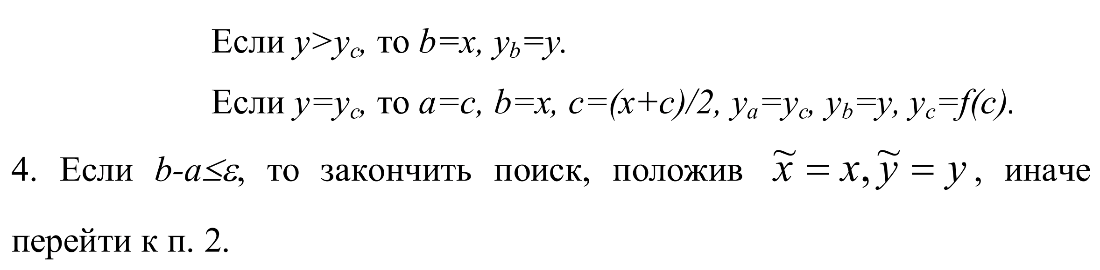




Метод парабол.

Алгоритм:





Код программы:

Файл function.mlx (наша функция по условиям):

function y = fun(x)

y = x .^2 + exp(x);

Файл solution.mlx (сборка всех файлов и запуск программы):

clear, clc

a = -1;

b = 0;

c = -0.5;

n = 18;

x1 = last('fun', a, b, c, n);

fm = fun(x1);

xm = x1;

fplot(@fun, [a, b]);

grid on

hold on

plot([a, b], fun(xm))

Файл last.mlx (фактически решение):

function x = func(fun, a, b, c, n)

x = 1;

for k=1:n

ya = feval(fun, a);

yc = feval(fun, c);

yb = feval(fun, b);

t = c + 0.5 \* (((b - c)^2) \*(ya - yc) - ((c - a)^ 2) ...

\* (yb - yc)) / ((b - c) \* (ya - yc) + (c - a) \* (yb - yc))

if (x < 0.5)

break;

end;

if t == c

x = (a + c) /2;l

else

x = t;

end

y = feval(fun, x);

if x < c

if y == yc

a = x, b = c, c = (x + c) / 2, ya = y,

yb = yc, yc = feval(fun, c);

else

if y > yc

a = x, ya = y

else

b = c, c = x, yb = yc, yc = y

end

end

else

if y == yc

a = c, b = x, c = (x + c) / 2, ya = y, yb = yc,

yb =y, yc = feval(fun, c);

else

if y > yc

b = x, yb = y

else

a = c, c = x, ya = yc, yc = y

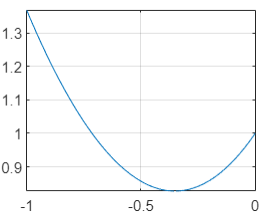
end

end

end

end

График, точка минимума и знaчение функции в ней:







Выводы:

|  |  |  |
| --- | --- | --- |
| Номер метода | Значение функции при минимуме | Точка минимума |
| 1 | 0.8276 | -0.3333 |
| 2 | 0.8272 | -0.3517 |
| 3 | 0.8297 | -0.3090 |
| 4 | 2.5601 | -1.5309 |
| 5 | 1 | -0.000038147 |
| 6 | 0.8565 | -0.5 |
| 7 | 0.8273 | -0.3595 |

Почти что все методы дали приблизительно правильный ответ. Исключение метод Фибоначчи. Связано это с тем, что алгоритм метода не подразумевает ограничение на аргумент, из-за чего последний вышел за пределы данного отрезка, как следствие значение функции не оптимальное.