|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ «Информатика и системы управления»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Лабораторная работа № \_\_**1**\_\_**

**Дисциплина Методы вычислений**

|  |  |
| --- | --- |
| **Тема Метод поразрядного поиска**  **Вариант №10**  **Студент \_Коноваленко В. Д.\_\_\_\_\_\_\_\_**  **Группа \_ИУ7-21М\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Оценка (баллы) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Преподаватель \_Власов П.А.** |  |

Москва.

2024 г.

**Цель работы:** изучение метода поразрядного поиска для решения задачи одномерной минимизации.

**Содержание работы**

1. реализовать метод поразрядного поиска в виде программы на ЭВМ;
2. провести решение задачи

для данных индивидуального варианта;

1. организовать вывод на экран графика целевой функции, найденной точки минимума и последовательности точек приближающих точку искомого минимума (для последовательности точек следует предусмотреть возможность «отключения» вывода её на экран).

|  |  |
| --- | --- |
| **Целевая функция *f(x)*** | ***[a, b]*** |
|  | *[0, 1]* |

Метод поразрядного поиска является усовершенствованием метода перебора с целью уменьшения числа обращений к целевой функции.

Основная идея: на начальном этапе, используя сравнительно большой шаг определяют примерную локализацию точки минимума, далее в полученной окрестности значение точки минимума уточняют с использованием более мелкого шага.

В основе метода лежит одно из свойств унимодальных функций:

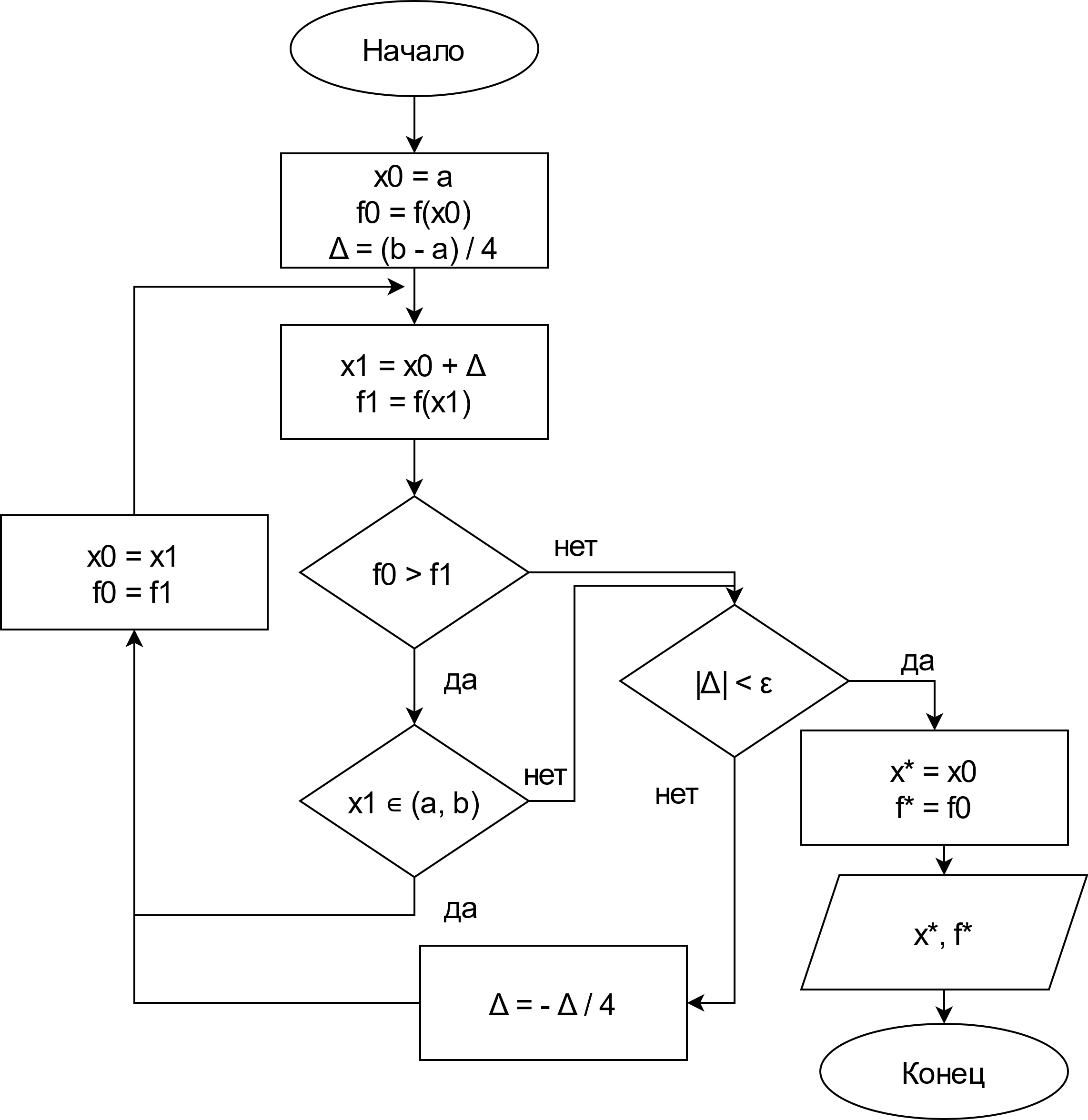
Если , то

=>

=>

Обычно начальное значение шага берут , затем вычисляют значения

до тех пор, пока на некотором шаге не будет выполнено условие: . В этих случаях направление поиска изменяют на противоположное и уменьшают шаг (как правило, также в 4 раза).



Текст программы представлен на Листинге 1

Листинг 1

|  |
| --- |
| function lab1()  debug = 1;  a = 0;  b = 1;  e = 0.01;  [x, y, N] = bitwiseSearch(a, b, e, debug);    fplot(@f, [a, b]);  hold on;  fprintf('RESULT: e = %f | N = %d | x\* = %.10f | f(x\*) = %.10f', e, N, x, y);  scatter(x, y, 'b', 'filled');  end  function y = f(x)  y = sin((power(x, 4) + power(x, 3) - 3 \* x + 3 - power(30, 1/3)) / 2) + tanh((4 \* sqrt(3) \* power(x, 3) - 2 \* x - 6 \* sqrt(2) + 1) / (-2 \* sqrt(3) \* power(x, 3) + x + 3 \* sqrt(2))) + 1.2;  end  function [x, y, N] = bitwiseSearch(a, b, e, debug)  i = 0;  x0 = a;  f0 = f(x0);  delta = (b - a) / 4;  if debug  hold on;  fprintf('%d: x%d = %.10f | y%d = %.10f\n', 0, 0, x0, 0, f0)  scatter(x0, f0);  end  while 1  i = i + 1;  x1 = x0 + delta;  f1 = f(x1);  if debug  fprintf('%d: x%d = %.10f | y%d = %.10f\n', i, i, x1, i, f1)  scatter(x1, f1);  end  if f0 <= f1 || x1 <= a || x1 >= b  if abs(delta) < e  break;  else  delta = -delta / 4;  end  end  x0 = x1;  f0 = f1;  end  x = x0;  y = f0;  N = i + 1;  end |

**Результаты расчетов для задачи из индивидуального варианта.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № п/п | ε | N |  |  |
| 1 | 0.01 | 21 | 0.7070312500 | -0.4652470295 |
| 2 | 0.0001 | 30 | 0.7054443359 | -0.4652516055 |
| 3 | 0.000001 | 47 | 0.7054662704 | -0.4652516064 |