|  |  |
| --- | --- |
| 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\_\_\_\_\_\_\_\_

**ОТЧЕТ**

**по Лабораторной работе №3**

**по курсу**

**«Методы вычислений»**

**Тема**

**«Метод парабол»**

**Вариант 9**

Студент \_\_\_ИУ7-21М\_\_\_\_  **\_\_\_**Карпухин А.С.\_\_\_

(Группа) (И.О.Фамилия)

Преподаватель  **\_\_\_\_**Власов П.А.**\_\_\_\_\_\_**

(И.О.Фамилия)

1. *г.*
2. **Постановка задачи**

Общий вид задачи оптимизации:

где , .

Если в данной задаче , то соответствующая задача называется задачей одномерной оптимизации.

Задача одномерной минимизации имеет вид:

1. **Исходные данные варианта**

Целевая функция *f*:

a = 0, b = 1.

1. **Метод парабол**

Метод парабол заключается в замене целевой функции на рассматриваемом отрезке полиномом второй степени и поиске точки минимума для данного полинома.

Изначально выбирается три точки , , , через которые проводится парабола. Точка ее минимума вычисляется по формуле:

|  |  |
| --- | --- |
|  | (1) |

где

Затем из полученных четырех точек выбирается три методом исключения отрезков на основе свойства унимодальных функций для следующей итерации.

Начальные точки , , выбираются так, чтобы удовлетворять условию:

|  |  |
| --- | --- |
|  | (2) |

Алгоритм метода парабол:

|  |
| --- |
| **Шаг 1**. Выбрать , , согласно условию (2).  **Шаг 2.** Найти по формуле (1). На первой итерации перейти к шагу 4. На остальных – к шагу 3.  **Шаг 3.** Проверка на окончание поиска: если длина отрезка или разность между предыдущим и текущим найденным значением меньше точности ε, то завершить поиск, полагая . Иначе перейти к шагу 4.  **Шаг 4.** Вычислить значение .  **Шаг 5.** Выбрать новую тройку , , для следующей итерации методом исключения отрезков и присвоить , , соответствующие значения выбранных точек. Перейти к шагу 2. |

Для выбора начальных точек предварительно используется метод золотого сечения, до тех пор, пока пробные точки метода, выбранные на очередной итерации, а также одна из крайних точек текущего отрезка не будут удовлетворять условию (2).

**Текст программы**

Код разработанной программной реализации метода золотого сечения приведен в листинге 1.

Листинг 1. Код программы, реализующей метод золотого сечения

|  |
| --- |
| function lab3()  f = @(x) (tan((x.^4 + 2.\*x.^2 - 2.\*x + 2^0.5 + 1) ./ 8) + ...  sin((4.\*x.^3 - 7.\*x - 9) ./ (20.\*x + 28)));    a = 0; b = 1;  X = a:0.01:b;  Y = f(X);  eps = [0.01, 0.0001, 0.000001];    fprintf("\nРезультаты вычисления точки минимума\n");  fprintf("для различных значений точности:\n\n");  fprintf("Eps - точность\n");  fprintf("N - число обращений к целевой функции\n");  fprintf("x\* - найденная точка минимума функции\n");  fprintf("f(x\*) - найденный минимум функции\n\n");  fprintf(" # | Eps | N | x\* | f(x\*)\n");  fprintf("---|----------|----|---------|--------\n");    figure('Units', 'normalized', 'OuterPosition', [0 0 1 1]);  title('Метод поразрядного поиска');  tiledlayout(2, 2);    for i = 1:length(eps)  % Вычисление точки минимума и минимума функции  [X1, X2, X3, A\_S, B\_S, N1] = GoldenRatio(a, b, f);  [X0, F0, L\_S, R\_S, N2] = ParabolaMethod(X1, X2, X3, f, eps(i));    A\_S = [A\_S, L\_S];  B\_S = [B\_S, R\_S];    % Вывод строки таблицы результатов вычислений  fprintf("%2i | %5f | %2i | %5.5f | %5.5f\n", ...  i, eps(i), N1 + N2, X0, F0);    % Вывод графика для данной точности  ax = nexttile;  % Целевая функция  plot(ax, X, Y, '-b','LineWidth',1.5);  hold on;  % Последовательность приближений  plot(ax, [A\_S, B\_S], [f(A\_S), f(B\_S)], '\*g','LineWidth', 2);  % Точка минимума  plot(ax, X0, F0, '\*r','LineWidth', 4);  title(ax, sprintf("Точность Eps = %2.0e", eps(i)));  legend('Целевая функция', 'Последовательность приближений', ...  'Точка минимума');  end    end    % Метод парабол  function [X, F, A\_S, B\_S, N] = ParabolaMethod(x1, x2, x3, f, eps)  arguments  x1 double % Первая начальная точка (левая граница отрезка)  x2 double % Вторая начальная точка (внутренняя точка отрезка)  x3 double % Третья начальная точка (правая граница отрезка)  f function\_handle % Целевая функция  eps double % Точность  end    A\_S = [];  B\_S = [];    f1 = f(x1); f2 = f(x2); f3 = f(x3);  x\_curr = x2;  f\_s = f2;  N = 3; % Число обращений к целевой функции    % Повторять, пока отрезок или разность между старым  % и новым значением f(x\*) не станет меньше точности  while true  A\_S = [A\_S, x1];  B\_S = [B\_S, x3];    r12 = x1^2 - x2^2;  r23 = x2^2 - x3^2;  r31 = x3^2 - x1^2;    s12 = x1 - x2;  s23 = x2 - x3;  s31 = x3 - x1;    % Вычисление точки минимума аппроксимирующей параболы  x\_prev = x\_curr;  x\_curr = 0.5 \* (f1\*r23 + f2\*r31 + f3\*r12) / (f1\*s23 + f2\*s31 + f3\*s12);  f\_s = f(x\_curr); N = N + 1;    % Проверка условия окончания поиска  if (x3 - x1) <= eps || abs(x\_prev - x\_curr) <= eps  break;  end    % Выбор тройки точек для следующей итерации  if x\_curr >= x2 && x\_curr <= x3  if f\_s <= f2  x1 = x2; x2 = x\_curr;  f1 = f2; f2 = f\_s;  else  x3 = x\_curr;  f3 = f\_s;  end  elseif x\_curr >= x1 && x\_curr <= x2  if f\_s <= f2  x3 = x2; x2 = x\_curr;  f3 = f2; f2 = f\_s;  else  x1 = x\_curr;  f1 = f\_s;  end  end  end    X = x\_curr;  F = f\_s;  end    % Метод золотого сечения  function [X1, X2, X3, A\_S, B\_S, N] = GoldenRatio(a, b, f)  arguments  a double % Левая граница отрезка  b double % Левая граница отрезка  f function\_handle % Целевая функция  end    tau = (5^0.5 - 1) / 2;    A\_S = [];  B\_S = [];    l = b - a;    % Вычисление пробных точек  x1 = b - tau \* l;  x2 = a + tau \* l;    f1 = f(x1);  f2 = f(x2);    N = 2;    while true  A\_S = [A\_S, a];  B\_S = [B\_S, b];    % Выбор новых точек для следующей итерации  if (f1 >= f2)  a = x1;  l = b - a;    x1 = x2;  f1 = f2;    x2 = a + tau \* l;  f2 = f(x2);  N = N + 1;  else  b = x2;  l = b - a;    x2 = x1;  f2 = f1;    x1 = b - tau \* l;  f1 = f(x1);  N = N + 1;  end    % Проверка условия выбора начальных точек для метода парабол  if (x1 < x2) && (x2 < b) && (f2 <= f1) && (f2 <= f(b))  X1 = x1;  X2 = x2;  X3 = b;  break;  elseif (a < x1) && (x1 < x2) && (f1 <= f(a)) && (f1 <= f2)  X1 = a;  X2 = x1;  X3 = x2;  break;  end  end  end |

1. **Результаты расчётов**

Результаты расчетов для задачи одномерной минимизации в соответствии с индивидуальным вариантом для различных значений *ε* приведены ниже в таблице 1.

Таблица 1. Результаты расчетов для задачи индивидуального варианта.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *№* | *ε* | *N* | *x\** | *f(x\*)* |
| 1 | 10-2 | 7 | 0.37497 | -0.06529 |
| 2 | 10-4 | 11 | 0.38378 | -0.06533 |
| 3 | 10-6 | 14 | 0.38379 | -0.06533 |