|  |  |
| --- | --- |
| 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 № \_\_**2**\_\_**

**Дисциплина Методы вычислений**

|  |  |
| --- | --- |
| **Тема Метод золотого сечения**  **Вариант №1**  **Студент \_Андреев А.А.\_\_\_\_\_\_\_\_**  **Группа \_ИУ7-22М\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Оценка (баллы) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Преподаватель \_Власов П.А.** |  |

Москва.

2024 г.

**Цель работы:** изучение метода золотого сечения для решения задачи одномерной минимизации.

**Содержание работы**

1. реализовать метод золотого сечения 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 отрезков содержащих точку искомого минимума (для последовательности отрезков следует предусмотреть возможность «отключения» вывода её на экран).

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

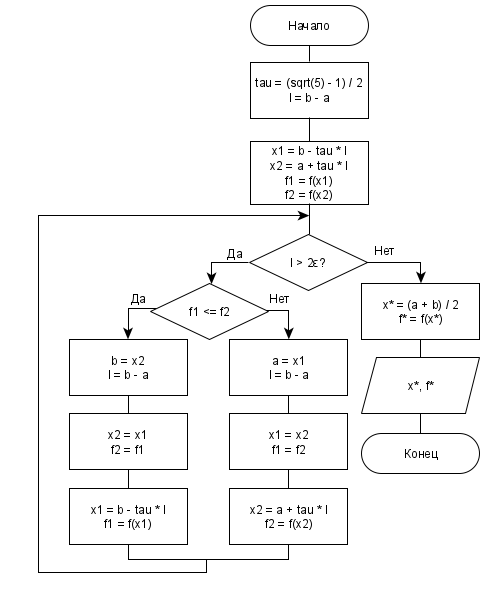
В основе метода золотого сечения лежит идея об уменьшении числа обращений к целевой функции засчёт того, что одна из пробных точек текущей итерации может быть использована и на следующей.

Пробные точки x1, x2 выбираются симметрично относительно середины отрезка [a, b] (это нужно для того, чтобы относительное уменьшение длины отрезка () при переходе к следующей итерации не зависела от того, какая часть отрезка выбрана). τ выбирается таким образом, чтобы пробная точка x1 с текущей итерации стала бы одной из пробных точек на следующей итерации.

Каждая из пробных точек x1, x2 делит отрезок [a, b] на две независимые части таким образом, что

Точки, обладающие этим свойством, называются точками золотого сечения отрезка [a, b].

На каждой итерации длина отрезка уменьшается в τ раз. Поэтому после выполнения n итерации длина текущего отрезка будет равна .



Текст программы представлен на Листинге 1

Листинг 1

|  |
| --- |
| 1. **function** **lab02**() 2. clc; % Очистка командного окна 3. debugFlg = 1; % Флаг отладки 4. delayS = 0.8; % Задержка для отладочной информации 5. a = 0; % Начальное значение интервала 6. b = 1; % Конечное значение интервала 7. eps = 1e-6; % Точность 8. x = linspace(a, b, 100); % Генерация точек для построения графика 9. y = arrayfun(@f, x); % Вычисление значений функции f для точек x 10. plot(x, y); % Построение графика функции f 11. hold on; % Удержание текущего графика на рисунке 12. pause(3); % Пауза 13. [xStar, fStar] = goldenRatio(a, b, eps, debugFlg, delayS); % Вызов функции оптимизации 14. plot(xStar, fStar, 'ro'); % Подсветка точки минимума 15. **end** 16. **function** [xStar, fStar] = **goldenRatio**(a, b, eps, debugFlg, delayS) 17. tau = (sqrt(5) - 1) / 2; % Золотое сечение 18. l = b - a; % Длина интервала 19. x1 = b - tau \* l; % Вычисление первой точки 20. x2 = a + tau \* l; % Вычисление второй точки 21. f1 = f(x1); % Вычисление значения функции в первой точке 22. f2 = f(x2); % Вычисление значения функции во второй точке 23. i = 0; % Счетчик итераций 24. **while** 1 % Бесконечный цикл 25. i = i + 1; % Увеличение счетчика итераций 26. **if** debugFlg 27. fprintf('№ %2d ai=%.10f bi=%.10f\n', i, a, b); % Вывод отладочной информации 28. plot([a b], [f(a) f(b)], 'b'); % Построение отрезка 29. hold on; % Удержание текущего графика на рисунке 30. pause(delayS); % Пауза 31. **end** 32. **if** l > 2 \* eps % Проверка условия останова 33. **if** f1 <= f2 34. b = x2; 35. l = b - a; 36. x2 = x1; 37. f2 = f1; 38. x1 = b - tau \* l; 39. f1 = f(x1); 40. **else** 41. a = x1; 42. l = b - a; 43. x1 = x2; 44. f1 = f2; 45. x2 = a + tau \* l; 46. f2 = f(x2); 47. **end** 48. **else** 49. xStar = (a + b) / 2; % Найденное приближение минимума 50. fStar = f(xStar); % Значение функции в найденном минимуме 51. **break** % Выход из цикла 52. **end** 53. **end** 54. i = i + 1; % Увеличение счетчика итераций 55. **if** debugFlg 56. fprintf('№ %2d ai=%.10f bi=%.10f\n', i, a, b); % Вывод отладочной информации 57. fprintf('RESULT: x\*=%.10f f(x\*)=%.10f\n', xStar, fStar); % Вывод результата 58. plot([a b], [f(a) f(b)], 'r'); % Подсветка отрезка 59. **end** 60. **end** 61. **function** **y** = **f**(x) 62. y = exp(((x^4) + (x^2) - x + sqrt(5)) / 5) + sinh((x^3 + 21 \* x + 9) / (21\*x + 6)) - 3.0; 63. %y = (x - 0.50)^4; % Определение функции 64. **end** |

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № п/п | ε | N |  |  |
| 1 | 0.01 | 11 | 0.5688837075 | -0.0104319066 |
| 2 | 0.0001 | 20 | 0.5713631534 | -0.0104403334 |
| 3 | 0.000001 | 30 | 0.5713159013 | -0.0104403366 |