



МИНОБРНАУКИ РОССИИ
федеральное государственное бюджетное образовательное
учреждение высшего образования
«Московский государственный технологический университет
«СТАНКИН» (ФГБОУ ВО «МГТУ «СТАНКИН»)

**Институт
информационных
технологий**

**Кафедра
информационных технологий
и вычислительных систем**

ОТЧЕТ О ВЫПОЛНЕНИИ
ИНДИВИДУАЛЬНОГО ЗАДАНИЯ ПО ДИСЦИПЛИНЕ
«Методы оптимизации»

СТУДЕНТА 2 КУРСА бакалавриата ГРУППЫ ИДБ-22-04
(уровень профессионального образования)

Макаров Андрей Олегович

НА ТЕМУ
«Методы одномерной оптимизации»

Направление: 09.03.01 Информатика и вычислительная техника
Профиль подготовки: «Разработка программных комплексов в рамках цифровой трансформации деятельности предприятий»

Отчет сдан « _____ » _____ 20 ____ г.

Оценка _____

Преподаватель Палванов М.Р.

(Ф.И.О., должность, степень, звание.)

(подпись)

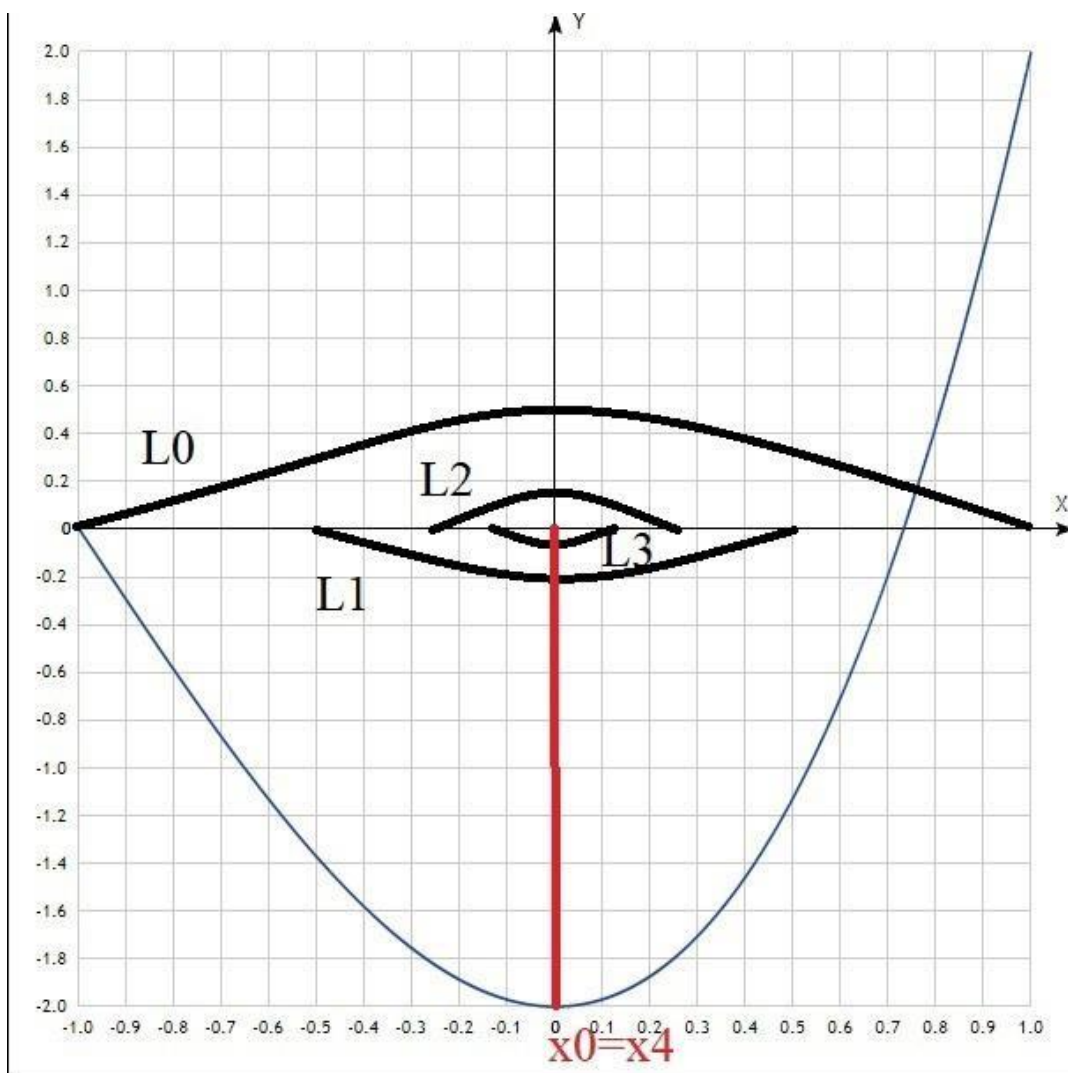
МОСКВА 2024

Метод половинного деления

Дано:

$$f(x) = x^3 - 3x^2 - 2 \quad L_0 = [-1; 1] \quad \varepsilon = 0,2$$

Минимум функции был сразу найден с помощью программного кода,
График функции из индивидуального задания и геометрическая интерпретация метода представлены ниже:



Код программы представлен на языке C++, метод половинного деления реализуется в строчках 10-60.

```
1 // NO Л1.cpp : Этот файл содержит функцию "main". Здесь начинается и заканчивается выполнение программы.
2 //
3 #define _CRT_SECURE_NO_WARNINGS
4
5 #include <stdio.h>
6 #include <math.h>
7 #include <iostream>
8 //Вариант 9 f(x)=x^3+3x^2-2 L0=[-1,1] e=0,2
9
10 int polovinadel(double x1, double x2, double x3, double a, double b, double e) {
11     printf("
12     printf("
13     double xs = (a + b) / 2, l = fabsf(b - a);
14
15
16     int k = 0;
17
18     m1:
19     k += 1;
20     double fx = x1 * xs * xs * xs + x2 * xs * xs + x3;
21     double y = a + l / 4, z = b - l / 4;
22     double fy = x1*y * y * y + x2*y * y + x3;
23     double fz = x1*z * z * z + x2*z * z + x3;
24     // printf("f(x) = %lf f(y) = %lf f(z) = %lf\n", fx, fy, fz);
25     if (fy < fx) {
26         printf("f(y)<f(x) "); // var 5
27         b = xs;
28         xs = y;
29     }
30     else {
31         if ((fz < fx) && (fy > fx)) {
32             printf("f(z)<f(x) "); // var 6
33             a = xs;
34             xs = z;
35             // printf("2\n");
36         }
37         else if (fz >= fx) {
38             printf("f(z)>=f(x) ");
39             a = y;
40             b = z;
41             // printf("3\n");
42         }
43     }
44 }
```

Листинг 1.

```
44 }
45
46 // var 7
47 l = fabsf(b - a);
48 if (l <= e) {
49     printf("ОТВЕТ \n");
50     printf("x -- [%f ; %f] x=%f f(x)=%f k= %d\n ", a, b, xs, x1 * xs * xs * xs + x2 * xs * xs + x3, k-1);
51 }
52 else {
53     printf("FROM RESULT ///////////////x -- [%f ; %f] x=%f f(x)=%f ||| var k= %d ||| y=%f z=%f f(y)=%f f(z)=%f ///////////////\n", a, b, xs, x1 * xs * xs * xs + x2 * xs * xs + x3, k, y, z, fy, fz);
54     goto m1;
55 }
56
57 printf("
58 return 0;
59 }
60
61
62
63
64
65 int zoloto(double x1, double x2, double x3, double a, double b, double e) {
66     printf("
67     printf("
68     double l = 0;
69     int k = 0;
70     double y = a + ((3 - sqrt(5)) / 2) * (b - a), z = a + b - y;
71
72     m1:
73     k += 1;
74     double fy = x1 * y * y * y + x2 * y * y + x3;
75     double fz = x1 * z * z * z + x2 * z * z + x3;
76
77     if (fy <= fz) {
78         printf("f(y)<=f(x) ");
79         b = z;
80         y = a + b - y;
81     }
82     else {
83         printf("f(y)>f(x) ");
84         a = y;
85         y = z;
86         z = a + b - z;
87     }
```

Листинг 2.

```

87     z = a + b - z;
88 }
89 l = fabsf(a - b);
90 if (l == 0) {
91     printf("OTBET \n");
92     printf("x == [%d.f ; %d.f]  x==%d.f  f(x)==%d.f  k = %d\n ", a, b, ((a+b)/2), x1 + ((a+b)/2) * ((a+b)/2) + x2 + ((a+b)/2) * ((a+b)/2) + x3, k-1);
93 }
94 else {
95     printf("FROM RESULT  ///////////x == [%d.f ; %d.f]  x==%d.f  f(x)==%d.f  |||  War k = %d  ||||  y=%d.f  z=%d.f  f(y)==%d.f  f(z)==%d.f/////////\n", a, b, ((a+b)/2), x1 + ((a+b)/2) * ((a+b)/2) + x3, k-1);
96     goto m1;
97 }
98
99 printf("-----\n");
100 return 0;
101 }
102
103
104
105 int fibonacci(double x1, double x2, double x3, double a, double b, double e) {
106     printf("-----\n");
107     printf("          МЕТОД ЧИСЕЛ ФИБОНАЧЧИ          \n");
108     m2:
109     printf("Введите дополнительно допустимую длину конечного интервала и константу различности, а также количество вычислений функции (3<N<10 000) для выполнения поиска методом чисел Фибоначчи\n");
110     double l;
111     int n;
112     scanf("%d %d", &l, &n);
113     if (n <= (fabsf(a + b) / l)) {
114         printf("Введите большее количество вычислений функции");
115         goto m2;
116     }
117     double fib[10000];
118
119     fib[1] = 1;
120     fib[2] = 1;
121     for (int i = 3; i < 10000; i++)
122         fib[i] = fib[i-1] + fib[i-2];
123     int k = 0;
124     double y = a + (fib[n-2] / fib[n]) * (b - a);
125     double z = a + (fib[n-1] / fib[n]) * (b - a);
126
127 m1:
128     double fy = x1 + y + y + x2 + y + y + x3;
129     double fz = x1 + z + z + x2 + z + z + x3;
130

```

Листинг 3.

```

130
131 if (fy <= fz) {
132     printf("f(y)<=f(z)  f(y)==%d.f  f(z)==%d.f  k = %d  \n", fy, fz, k);
133     b = z;
134     z = y;
135     y = a + (fib[n - k - 2] / fib[n - k - 1]) * (b - a);
136 }
137 else
138     if (fy > fz)
139     {
140         printf("f(y)>f(z)  f(y)==%d.f  f(z)==%d.f  k = %d  \n", fy, fz, k);
141         a = y;
142         y = z;
143         z = a + (fib[n - k - 2] / fib[n - k - 1]) * (b - a);
144     }
145
146
147 if (k != (n - 3)) {
148     k = k + 1;
149     // printf("k = %d\n ", k);
150     goto m1;
151 }
152 else
153     if (k == (n - 3)) {
154         y = z + (a + b) / 2;
155         z = y + e;
156         //printf("y and z = %d.f ||| %d.f, y, z);
157         //printf("a and b = %d.f ||| %d.f, a, b);
158         fy = x1 + y + y + x2 + y + y + x3;
159         fz = x1 + z + z + x2 + z + z + x3;
160
161         if (fy <= fz) {
162             b = z;
163         }
164         else
165         {
166             a = y;
167         }
168         // printf("a and b = %d.f ||| %d.f, a, b);
169         // printf("\n");
170         printf("\n");
171         printf("\n");
172         printf("\n");
173     }

```

Листинг 4.

```

173     printf("\n");
174     printf("\n");
175     printf("OTBET \n");
176     printf("x == [%d.f ; %d.f]  x==%d.f  f(x)==%d.f\n ", a, b, ((a+b)/2), x1 + ((a+b)/2) * ((a+b)/2) + x2 + ((a+b)/2) * ((a+b)/2) + x3);
177     printf("-----\n");
178 }
179
180 return 0;
181 }
182
183
184 int main()
185 {
186     setlocale(LC_ALL, "Russian");
187
188     double a, b, e;
189     double x1, x2, x3;
190     // printf("Введите коэф.сначала перед x^3 затем перед x^2 , затем свободный член\n");
191     //scanf("%d %d %d", &x1, &x2, &x3);
192     //printf("Введите l0([a;b]) и точность\n");
193     //scanf("%d %d", &a, &b);
194     printf("Введите коэффициент перед x^3, затем перед x^2, затем свободный член.\n");
195     scanf("%d %d %d", &x1, &x2, &x3);
196     printf("коэффициенты: (%d.f)^3+(%d.f)^2+(%d.f)\n", x1, x2, x3);
197     printf("Введите начало и конец интервала l0([a;b]) и точность.\n");
198     scanf("%d %d %d", &a, &b, &e);
199     printf("Интервал: [%d.f ; %d.f], Точность: %d.f\n", a, b, e);
200     printf("\n");
201     printf("\n");
202     printf("\n");
203     polovinadel(x1, x2, x3, a, b, e);
204     zoloto(x1, x2, x3, a, b, e);
205     fibonacci(x1, x2, x3, a, b, e);
206 }

```

Листинг 5.

Ответ программы с подробным описанием:

```

Введите коэффициент перед x^3, затем перед x^2, затем свободный член:
3
-2
Коэффициенты: (1,000000)x^3+(3,000000)x^2+(-2,000000)
Введите начало и конец интервала L0([a;b]) и точность:
-1
1
0,2
Интервал: [-1,000000; 1,000000], Точность: 0,200000

-----
МЕТОД ПОЛОВОИННОГО ДЕЛЕНИЯ
f(z)>f(x)  PROM RESULT  //x -- [-0,500000 ; 0,500000]  x*=-0,000000  f(x)=-2,000000  |||  Шаг k = 1  |||  y=-0,500000 z=0,500000 f(y)=-1,375000  f(z)=-1,125000  //x
f(z)>f(x)  PROM RESULT  //x -- [-0,250000 ; 0,250000]  x*=-0,000000  f(x)=-2,000000  |||  Шаг k = 2  |||  y=-0,250000 z=0,250000 f(y)=-1,828125  f(z)=-1,796875  //x
f(z)>f(x)  PROM RESULT  //x -- [-0,125000 ; 0,125000]  x*=-0,000000  f(x)=-2,000000  |||  Шаг k = 3  |||  y=-0,125000 z=0,125000 f(y)=-1,955078  f(z)=-1,951172  //x
f(z)>f(x)  OTBET
x -- [-0 ; 0]  x*=-0  f(x)=-2 k= 3
-----
МЕТОД ЗОЛОТОГО СЕЧЕНИЯ
f(y)<f(x)  PROM RESULT  //x -- [-1,000000 ; 0,236068]  x*=-0,381966  f(x)=-1,618034  |||  Шаг k = 1  ||||  y=-0,527864 z=-0,236068 f(y)=-1,845971  f(z)=-1,819660  //x
f(y)>f(x)  PROM RESULT  //x -- [-0,527864 ; 0,236068]  x*=-0,145898  f(x)=-1,939247  |||  Шаг k = 2  ||||  y=-0,236068 z=-0,055728 f(y)=-1,311163  f(z)=-1,845971  //x
f(y)>f(x)  PROM RESULT  //x -- [-0,236068 ; 0,236068]  x*=-0,000000  f(x)=-2,000000  |||  Шаг k = 3  ||||  y=-0,055728 z=0,055728 f(y)=-1,845971  f(z)=-1,998856  //x
f(y)<f(x)  PROM RESULT  //x -- [-0,236068 ; 0,055728]  x*=-0,098170  f(x)=-1,976341  |||  Шаг k = 4  ||||  y=-0,124612 z=-0,055728 f(y)=-1,998856  f(z)=-1,990510  //x
f(y)>f(x)  OTBET
x -- [-0 ; 0]  x*=-0  f(x)=-2 k= 4
-----
МЕТОД ЧИСЕЛ ФИБОНАЧЧИ
Введите дополнительно допустимую длину конечного интервала и косинтну различности, а также количество вычислений функции (3<N<10 000) для выполнения поиска методом чисел Фибоначчи
0,2
0,2
8
f(y)<f(z)  f(y)=-1,8434291  f(z)=-1,8164351  k = 0
f(y)>f(z)  f(y)=-1,3205921  f(z)=-1,8434291  k = 1
f(y)>f(z)  f(y)=-1,8434291  f(z)=-1,9933051  k = 2
f(y)<f(z)  f(y)=-1,9933051  f(z)=-1,9930891  k = 3
f(y)>f(z)  f(y)=-1,9416911  f(z)=-1,9933051  k = 4
f(y)<f(z)  f(y)=-1,9933051  f(z)=-1,9933051  k = 5

OTBET
x -- [-0,142857 ; 0,104762]  x*=-0,019048  f(x)=-1,998918

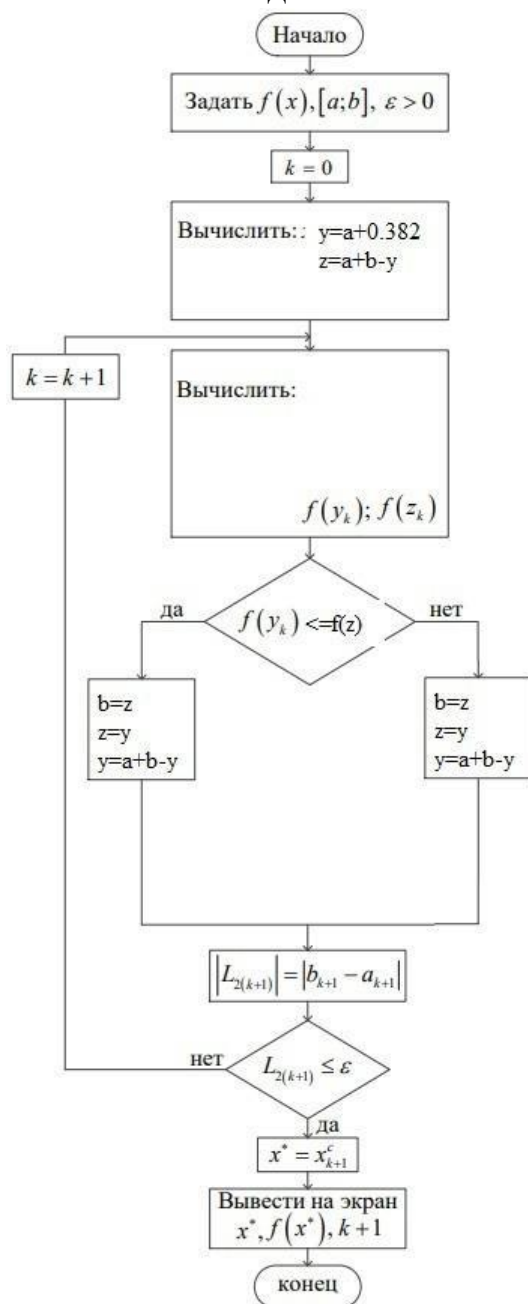
```

Как видно, программа не зависит от номера варианта, но в данном случае
 решается именно 9.

Для МПД $x \in [0; 0]$ $x^* = 0$ $f(x) = -2$ $k = 3$

Метод «золотого» сечения

Блок схема метода:



Код метода реализуется с 65 по 101 строчку.

Ответ программы для МЗС: $x \in [0; 0]$ $x^* = 0$ $f(x) = -2$ $k = 3$

Метод чисел Фибоначчи

Метод чисел Фибоначчи реализован со 105 по 182 строчку программного кода.

А также дополнительно из основной программы пользователю требуется ввести дополнительную информацию, а именно: допустимую длину конечного интервала, константу различимости, количество вычислений функции.

Для МФЧ:

$$x \in [-0,142857 ; 0,104762] \quad x^* = -0,019048 \quad f(x) = -1,998918 \quad k = 5$$

Заключение

Из 3 методов самым лучшим для этого варианта оказался метод половинного деления, потому что так как минимум этой функции на отрезке $[-1;1]$ находится ровно по середине, при $x = 0$, то этот метод в первой же итерации уткнулся в правильный ответ, не будь это так, результат был бы иным.