ЛАБОРАТОРНАЯ РАБОТА № 3

Метод парабол

Студент: Сергей Рамирович Диас Баскаков

Группа: ИУ7И-21М

Вариант № 22

1.

a. Постановка задачи:

$$\begin{cases} f(x) \to min \\ x \in [a; b] \end{cases}$$

b. Входные данные индивидуального варианта:

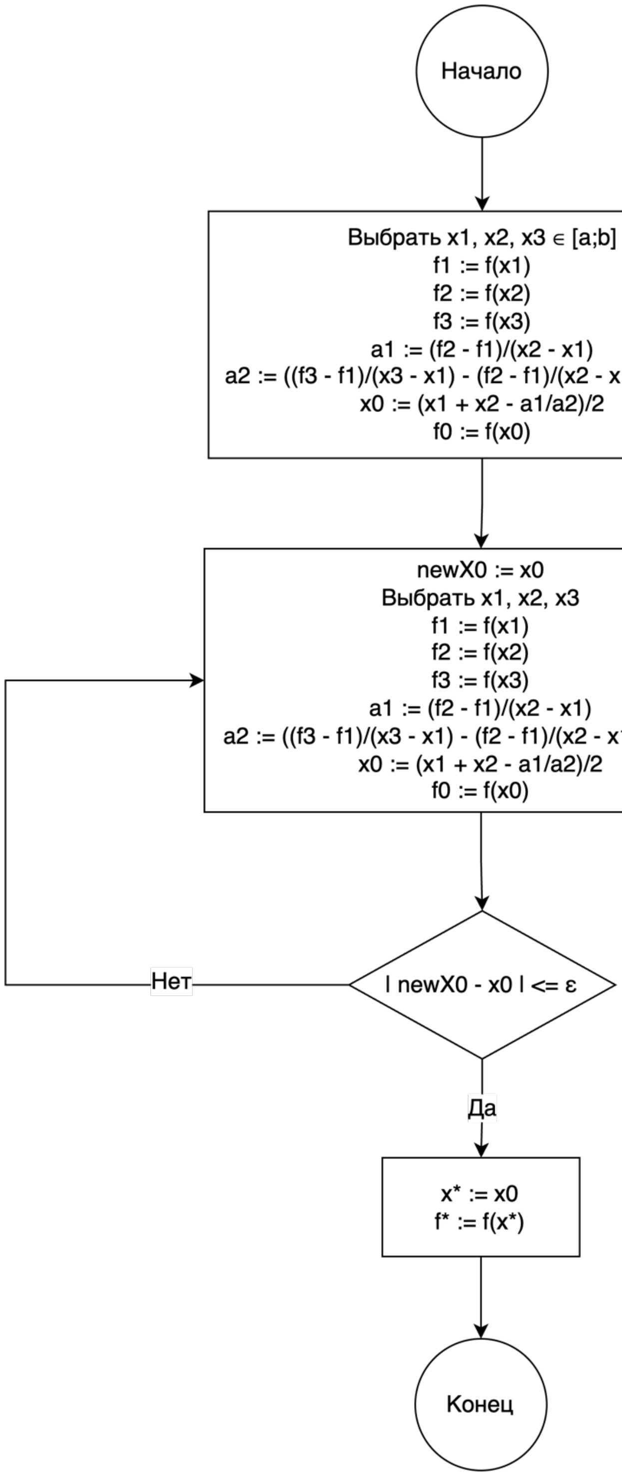
22	$\lg\left(-\sqrt{3}x^4 - x^2 + 5x + 1\right) + \operatorname{th}\left(\frac{-x^5 - 2x^4 - x^3 + 3x^2 + 6x + 3 - \sqrt{5}}{x^2 + 2x + 1}\right) - 1.0$	$[0, 1]$

2. Краткое описание метода парабол:
Этот метод используется для поиска минимума функции на отрезке. Он аппроксимирует функцию квадратичной параболой, используя три точки (x1, x2, x3), и находит минимум этой параболы. Затем он сужает интервал поиска, основываясь на информации о минимуме параболы. Процесс повторяется в новом интервале до достижения нужной точности.

Выбор точек x1, x2, x3:

- Точки должны удовлетворять следующим условиям:
  $x_1 < x_2 < x_3$
  $f(x_1) \geqslant f(x_2) \leqslant f(x_3)$
- На первом этапе часто достаточно использовать несколько пробных точек для определения x1, x2, x3. Если это занимает слишком много времени, можно выполнить несколько итераций метода золотого сечения до тех пор, пока пробные точки этого метода и одна из граничных точек текущего отрезка не удовлетворят вышеописанным условиям.
- На последующих этапах рассматриваются две пробные точки x2 и $\overline{x}$ на отрезке [x1, x3], используя метод исключения отрезков. В новом отрезке [x1’, x3’] в качестве x2’ выбирается точка x2 или $\overline{x}$, попавшая внутрь интервала.

Алгоритм:



3. Текст программы:

```
function [x_1, x_2, x_3] = choseInitialX(a, b, f)
    if ~isa(f, 'function_handle')
        error('Input f must be a function handle');
    end

    r = (sqrt(5) - 1) / 2;
    l = b - a;

    x1 = b - r*l;
```

```
    x2 = a + r*l;
    f1 = f(x1);
    f2 = f(x2);

    fa = f(a);
    fb = f(b);

    while true
        if ((a < x1) && (x1 < x2)) && ((fa >= f1) && (f1 <= f2))
            x_1 = a;
            x_2 = x1;
            x_3 = x2;
            break;
        elseif ((x1 < x2) && (x2 < b)) && ((f1 >= f2) && (f2 <= fb))
            x_1 = x1;
            x_2 = x2;
            x_3 = b;
            break;
        end

        if f1 <= f2
            b = x2;
            l = b - a;

            fb = f2;

            x2 = x1;
            f2 = f1;

            x1 = b - r*l;
            f1 = f(x1);

        else
            a = x1;
            l = b - a;

            fa = f1;

            x1 = x2;
            f1 = f2;

            x2 = a + r*l;
            f2 = f(x2);
        end
    end
end

function [result, count] = minimizationMethod(a, b, e, f, showPoints)
    % Check if f is a function handle
    if ~isa(f, 'function_handle')
        error('Input f must be a function handle');
    end

    [x1, x2, x3] = choseInitialX(a, b, f);

    f1 = f(x1);
    f2 = f(x2);
    f3 = f(x3);

    a1 = (f2 - f1)/(x2 - x1);
    a2 = ((f3 - f1)/(x3 - x1) - (f2 - f1)/(x2 - x1))/(x3 - x2);
    x0 = (x1 + x2 - a1/a2)/2;
```

```matlab
    f0 = f(x0);

    count = 4;

    if showPoints
        disp([num2str(x1, 6), ' , ', num2str(x3, 6)]);
    end
    while true
        newX0 = x0;

        if x0 < x2
            if f0 <= f2
                x3 = x2;
                x2 = x0;
                f3 = f2;
                f2 = f0;
            else
                x1 = x0;
                f1 = f0;
            end
        else
            if f2 <= f0
                x3 = x0;
                f3 = f0;
            else
                x1 = x2;
                x2 = x0;
                f1 = f2;
                f2 = f0;
            end
        end

        a1 = (f2 - f1)/(x2 - x1);
        a2 = ((f3 - f1)/(x3 - x1) - (f2 - f1)/(x2 - x1))/(x3 - x2);
        x0 = (x1 + x2 - a1/a2)/2;
        f0 = f(x0);

        count = count + 1;

        if showPoints
            disp([num2str(x1, 6), ' , ', num2str(x3, 6)]);
        end

        if abs(x0 - newX0) <= e
            result = x0;
            break;
        end
    end
end

% Define your function f(x)
f = @(x) -(log10(-sqrt(3)*x.^4 - x.^2 + 5*x + 1) + tanh((-x.^5 - 2*x.^4
- x.^3 + 3*x.^2 + 6*x + 3 - sqrt(5)) / (x.^2 + 2*x + 1)) - 1);

% Define a, b and e
a = 0;
b = 1;
e = 0.01;
%e = 0.0001;
%e = 0.000001;
```

```
% Call the minimizationMethod with the parameters and the function
handle
[result, count] = minimizationMethod(a, b, e, f, true);

% Display the result
disp(['x*: ', num2str(result, 6)]);
disp(['f*: ', num2str(f(result), 6)]);
disp(['N: ', num2str(count)]);
```

4. Результаты расчетов для задачи из индивидуального варианта:

No п/п	ε	N	x*	f(x*)
1	10^{-2}	7	0.741366	-0.512609
2	10^{-4}	12	0.744863	-0.512623
3	10^{-6}	18	0.744925	-0.512623