

## Задача 2. (Метод хорд)

Идея метода хорд состоит в том, что можно с известным приближением допустить, что функция на достаточно малом участке  $[a,b]$  изменяется линейно. Тогда кривую  $f(x)$  на участке  $[a,b]$  можно заменить хордой и в качестве приближенного значения корня принять точку пересечения хорды с осью абсцисс

$$x_1 = b - \frac{f(b) * (b - a)}{f(b) - f(a)}$$

Это значение аргумента используется для определения значения функции  $f(x_1)$ , которое сравнивается со значениями функций  $f(a)$  и  $f(b)$  и в дальнейшем используется вместо того из них, с которым оно совпадает по знаку.

Проиллюстрируем метод на примере поиска корня уравнения  $f(x) = \sin(x) - 1/x = 0$ .

Поиск решения в GNU Octave

Найдем интервал  $[a,b]$ , на котором функция  $f(x)$  меняет знак.

```
function y = f2(x)    % Функция сохранена в файле f2.m
y = sin(x) - 1./x ;
endfunction
% Программа поиска корня сохранена в файле (окно «Редактор»)

% Нарисуем график y(x)= sin(x)-1/x для определения
%% области нахождения корня [a,b]
%% Функция y(x)= sin(x)-1/x в файле с именем f2
cla;                % Очистка графических объектов
okno1=figure();    % Создание окна с идентификатором okno1
a=pi*0.5;          % Задаем начальный интервал нахождения корня
b=pi;
format short      % Задание формата вывода чисел
printf("  There are a= %d \n ", a) % Для контроля
printf("  There are b= %d \n ", b)
yy1 = f2(a)
yy2 = f2(b)
if (yy1*yy2) > 0 % Контроль интервала
    printf("  Mistake [a. b] \n ");
    return;
endif
x= pi()*0.01:0.2:pi();% Создание массива переменной x
y=f2(x);           % Создание массива переменной y
pol=plot(x,y);     % Создание графика
set(pol,'LineWidth',3,'Color','k'); % Установка толщины и цвета линии
set(gca,'xlim',[a,b]); % Установка осей
set(gca,'ylim',[-1,1]);
set(gca,'xtick',[pi*0.5:0.5:pi]); % Вывод сетки с заданным диапазоном
set(gca,'ytick',[-1:0.5:1]);
grid on;          % Установка сетки
xlabel('x');ylabel('y'); % Подписи осей
title('Plot y=sin(x) - 1/x'); % подпись графика
pause(2);
delete(okno1);
```

```

    % Из графика уточняем область нахождения корня [2.5; 3]
    eps=1.0e-7;          % Задаем точность решения
    af = 2.5;
    bf = 3;
    a =2.5;
    b =3 ;
    niter = 10;
    for i=1:niter % Поиск корня
        xx = b - (f2(b)*(b-a))/(f2(b)-f2(a));
        if (f2(xx)*f2(a) ) > 0
            a = xx;
        else
            b=xx;
        endif
        nev = abs(f2(xx)); % Невязка решения уравнения (значение функции)
        if (nev < eps) %% в точке корня
            printf(" Root are    xx = %d \n ", xx ); % Для контроля
            printf(" Discrepancy are nev = %d \n ", nev);
            % Формируем график функции в окрестности решения
            cla; % Очистка графических объектов
            okno2=figure(); % Создание окна с идентификатором okno2
            x= af:0.01:bf; % Создание массива переменной x
            y=f2(x); % Создание массива переменной y
            pol=plot(x,y); % Создание графика
            set(pol,'LineWidth',3,'Color','k') % Установка толщины и цвета линии
            set(gca,'xlim',[af,bf]); % Установка осей
            set(gca,'ylim',[-1 ,1 ]);
            set(gca,'xtick',[af:0.25:bf]); % Вывод сетки с заданным диапазоном
            set(gca,'ytick',[-1 :0.25:1]);
            grid on; % Установка сетки
            xlabel('x');ylabel('y'); % Подписи осей
            title('Rezult'); % подпись графика
            break;
        endif
    end
    printf(" Solution not found niter = %d", niter );

```

Результат работы программы в командном окне

```

    There are a= 1.5708
    There are b= 3.14159
    yy1 = 0.3634
    yy2 = -0.3183
    Root are    xx = 2.7726
    Discrepancy are nev = 9.9177e-08

```

Пояснения к программе. Программа состоит из двух блоков печати графиков, иллюстрирующих нахождение корня и блока непосредственного поиска корня (выделен цветом). Результат работы программы показывает, что корень уравнения  $\sin(x) - 1/x = 0$  есть 2.7726. Невязка решения - 9.9177e-08

Решение на C++

Листинг программы решения уравнения методом хорд представлен ниже.

```
#include <stdio.h>
#include <iostream>
#include <math.h>
const double PI = 3.14159265;
double fx(double x)
{
    double ff;
    ff = sin(x) - 1.0/x;
    return(ff);
}
int main()
{
    const double PI = 3.14159265;
    std::cout << "      Chord method \n";
    std::cout << "      The equation sin(x) - 1.0/x = 0 \n";
    double a = 3.1, b = PI/2.0;
    double eps = 1.0E-7;
    int i, niter = 100;
    if (fx(a) * fx(b) > 0)      printf("\t Mistake a= %f  b= %f \n", a, b);
    else {
        double xx, nev;
        for (i = 1; i < niter; i++) {
            xx = b - (fx(b)*(b - a))/(fx(b)-fx(a));
            if (fx(a) * fx(xx) > 0) a = xx; else b = xx;
            // printf("\t i=%i  a= %f  b= %f  x = %f  f(a)= %f  f(b)= %f \n", i, a, b, xx,
fx(a), fx(b));
            nev = fabs(fx(xx));
            if (nev < eps) {
                printf("\t  x = %f  nev = %e  iter= %i \n", xx, nev, i);
                break;}
            }
        }
        return 0;
    }
}
```

Листинг с результатом решения –

```
Chord method
The equation sin(x) - 1.0/x = 0
x = 2.772605    nev = 3.127263e-08    iter= 7
```