Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский Авиационный Институт»
(Национальный Исследовательский Университет)

Факультет прикладной математики и физики

Курсовой проект
по курсу «Вычислительные системы»
1 семестр

Задание 4
Процедуры и функции в качестве параметров

Руководитель
Довженко А.А.

(подпись)
(дата)
Студент
Группа М8О-113Б-21
Соломатина С.В.

(подпись)
(дата)

(оценка)

Москва, 2021г

Содержание

Введение

В данном задании курсового проекта нужно составить программу на языке Си с процедурами решения трансцендентных алгебраических уравнений различными численными методами. Нелинейные уравнения оформить как параметры-функции. Применить каждую процедуру к решению двух уравнений — варианта 20 и 21.

Методы решения трансцендентных алгебраических уравнений

Метод дихотомии

Метод дихотомии, или метод половинного деления, заключается в делении отрезка $[a;b]$ пополам и его сужении в два раза на каждом шаге итерационного процесса в зависимости от знака выражения $f(a)\cdot f(b)$. Это связано с тем, что, если на данном отрезке $[a;b]$ существует хотя бы один корень уравнения, то значения функции на концах этого отрезка имеют разные знаки: $f(a)\cdot f(b)<0$. Условием остановы итерационного процесса будет выполнение неравенства $|a-b|<\varepsilon$. Новый интервал вычисляется по формулам $a_k=(a_{k-1}+b_{k-1})/2$, $b_k=b_{k-1}$, если $f(a_{k-1})\cdot f((a_{k-1}+b_{k-1})/2)>0$; или по формулам $a_k=a_{k-1}$, $b_k=(a_{k-1}+b_{k-1})/2$, если $f(b_{k-1})\cdot f((a_{k-1}+b_{k-1})/2)>0$. Приближенное значение корня к моменту окончания итерационного процесса вычисляется по формуле $x_*=(a_{конечное}+b_{конечное})/2$.

Метод простых итераций

Идея метода итераций заключается в замене исходного уравнения $F(x)=0$ равносильным уравнением вида $x=f(x)$ с выделенным линейным членом.
Достаточное условие сходимости метода: $|f'(x)|<1, x\in[a,b]$. Это условие необходимо проверить перед началом решения задачи, так как функция $f(x)$ может быть выбрана неоднозначно, причём в случае неверного выбора указанной функции метод расходится.
Начальное приближение корня вычисляется по формуле: $x_0=(a+b)/2$;итерационный процесс задаёт формула $x_{k+1}=f(x_k)$, условие его окончания: $|x_k-x_{k-1}|<\varepsilon$.

Метод Ньютона

Метод Ньютона, или метод касательных, заключается в том, что если x_n — некоторое приближение к корню x_* уравнения $f(x)=0, f\in C^1$, то следующее приближение определяется как корень касательной к функции $f(x)$, проведённой в точке x_n. Найдём формулу для нахождения следующего приближения методом Ньютона из исходного уравнения $f(x_k)+f'(x_k)(x-x_k)=0$:

- Уравнение касательной имеет вид $f'(x_n)=\dfrac{f(x_n)-f(x_{n-1})}{x_n-x_{n-1}}$;

- Положим $y = 0$.
- Тогда $x_n = x_{n-1} - \dfrac{f(x_{n-1})}{f'(x_{n-1})}$.

Кроме того, для начала вычислений требуется задание начального приближения x_0, удовлетворяющего условию сходимости метода к корню x_*: $f(x_0) f''(x_0) > 0$.

Вычисления производятся, пока $|x_0 - x_n| > \varepsilon$.

Общий метод решения

Решить каждое из уравнений методом дихотомии, итераций, Ньютона или хорд. Передать в соответствующий метод в качестве аргументов исходную функцию (в случае метода Ньютона передать также производную) и отрезок, на котором будет производиться поиск корня.

Выразим для каждого варианта заданий функций вида *f(x) = x* для метода итераций и найдём производные для их использования в методе Ньютона.

Вариант 20

Требуется найти на отрезке [1; 2] приближенное значение трансцендентного уравнения $0{,}1x^2 - x\ln x = 0$. Дано приближенное значение корня 1.1183.

В нашем случае равносильное уравнение: $x = e^{0.1*x}$; имеем функцию $f(x) = 0{,}1x^2 - x\ln x$. Её первая производная равна $f'(x) = 0{,}2x - \ln - 1$, а вторая производная равна $f''(x) = 0{,}2 - \frac{1}{x}$. Возьмём машинное значение эпсилон.

Вариант 21

Требуется найти на отрезке $[0;0.8]$ приближенное значение трансцендентного уравнения $tgx - \frac{1}{3}tg^3x + \frac{1}{5}tg^5x - \frac{1}{3} = 0$. Дано приближенное значение корня 0.3333.

В нашем случае равносильное уравнение: $x = arctg(\frac{tg^3x}{3} - \frac{tg^5x}{5} + \frac{1}{3})$; имеем функцию $f(x) = tgx - \frac{1}{3}tg^3x + \frac{1}{5}tg^5x - \frac{1}{3}$. Её первая производная равна $f'(x) = \frac{1}{\cos^2 x} - \frac{\sin^2 x}{\cos^4 x} + \frac{\sin^4 x}{\cos^6 x} - \frac{1}{3}$, а вторая производная равна $f''(x) = \frac{2}{\cos^3 x} - \frac{2\sin x + \sin^3 x}{\cos^5 x} + \frac{4\sin^3 x + \sin^5 x}{\cos^7 x} - \frac{1}{3}$. Возьмём машинное значение эпсилон.

Функциональное значение

Программа предназначена для решение трансцендентных алгебраических уравнений различными численными методами (итераций, Ньютона и дихотомии). Значения ограничены переменными типа *double*.

Описание логической структуры

В необходимый метод поиска корня на отрезке подаётся сама функция, у которой нужно найти корень и значения концов самого отрезка. Далее в зависимости от самого метода осуществляется поиск корня до тех пор, пока погрешность измерений не будет меньше эпсилон, либо пока корень не будет найден точно.

Описание переменных и констант

Переменная	**Тип**	**Значение**
eps	double	7.0 / 3.0 - 4.0 / 3.0 - 1.0
e		2.718281828459045
a1		Нижняя граница отрезка для уравнения варианта 20
a2		Нижняя граница отрезка для уравнения варианта 21
b1		Верхняя граница отрезка для уравнения варианта 20
b2		Верхняя граница отрезка для уравнения варианта 21
x, x0, xn		Вспомогательные переменные для методов итераций, Ньютона, дихотомии

Описание функций

Функция	**Тип**	**Назначение**
f1v1	double	Функция из 20 варианта в исходном виде
f2v1		Функция из 21 варианта в исходном виде
f1v2		Функция из 20 варианта в виде $f(x)=x$
f2v2		Функция из 21 варианта в виде $f(x)=x$
f1d1		Первая производная функции из 20 варианта
f2d1		Первая производная функции из 21 варианта
f1d2		Вторая производная функции из 20 варианта
f2d2		Вторая производная функции из 21 варианта
dichotomy		Нахождение корня методом дихотомии
iteration		Нахождение корня методом итераций
Newton		Нахождение корня методом Ньютона

Протокол

```
#include <stdio.h>
#include <math.h>

#define eps (7.0 / 3.0 - 4.0 / 3.0 - 1.0)

const double e = 2.718281828459045;
const double a1 = 1, b1 = 2, a2 = 0, b2 = 0.8;

double f1v1(double x)
{
   return 0.1 * x * x - x * log(x);
}

double f1v2(double x)
{
   return pow(e, 0.1 * x) ;
}

double f2v1(double x)
{
   return tan(x) - pow(tan(x), 3) / 3 + pow(tan(x), 5) / 5 - 1.0 / 3;
}

double f2v2(double x)
{
   return atan(pow(tan(x), 3) / 3 - pow(tan(x), 5) / 5 + 1.0 / 3);
}

double f1d1(double x)
{
   return 0.2 * x - log(x) - 1;
}

double f1d2(double x)
{
   return 0.2 - 1 / x;
}

double f2d1(double x)
{
   return 1 / pow(cos(x), 2) - pow(sin(x), 2) / pow(cos(x), 4) + pow(sin(x), 4) / pow(cos(x), 6) - 1.0 / 3;
}

double f2d2(double x)
{
   return 2 / pow(cos(x), 3) - (2 * sin(x) + pow(sin(x), 3)) / pow(cos(x), 5) + (4 * pow(sin(x), 3) +
pow(sin(x), 5)) / pow(
```

```
        cos(x), 7) - 1.0 / 3;
}

double dichotomy(double f(double), double a, double b)
{
   while (fabs(a - b) > eps) {
      if (f(a) * f((a + b) / 2) > 0) {
         a = (a + b) / 2;
      }
      if (f(b) * f((a + b) / 2) > 0) {
         b = (a + b) / 2;
      }
   }

   return (a + b) / 2;
}

double iteration(double f(double), double a, double b)
{
   double x0 = (a + b) / 2;
   double xn = x0 + 1;
   while (fabs(x0 - xn) > eps) {
      xn = x0;
      x0 = f(x0);
   }
   return x0;
}

double newton(double f(double), double d1(double), double d2(double), double a, double b)
{
   double x0, xn;
   if (f(a) * d2(a) > 0) {
      x0 = a;
   } else {
      x0 = b;
   }

   xn = x0 - f(x0) / d1(x0);
   while (fabs(x0 - xn) > eps) {
      x0 = xn;
      xn = x0 - f(x0) / d1(x0);
   }
   return xn;
}

int main(void) {

   printf("|Вариант|\t|Метод дихотомии |\t|Метод итераций |\t|Метод Ньютона |\n");
```

```
    printf("|  20\t|\t");
    printf("|%lf\t\t |\t", dichotomy(f1v1, a1, b1));
    printf("|%lf\t\t|\t", iteration(f1v2, a1, b1));
    printf("|%lf\t   |\t\n", newton(f1v1, f1d1, f1d2, a1, b1));

    printf("|  21\t|\t");
    printf("|%lf\t\t |\t", dichotomy(f2v1, a2, b2));
    printf("|%lf\t\t|\t", iteration(f2v2, a2, b2));
    printf("|%lf\t   |\t\n", newton(f2v1, f2d1, f2d2, a2, b2));

    return 0;
}
```

Вывод программы:

Вариант	Метод дихотомии	Метод итераций	Метод Ньютона
20	1.118326	1.118326	1.118326
21	0.333255	0.333255	0.333255

Заключение

Задачи решения уравнений часто возникают на практике, и итерационные методы позволяют найти корень функции уравнения, заданной на отрезке.

Простейшим методом нахождения корней уравнений является метод деления пополам. К его достоинствам следует отнести высокую надёжность и простоту, к недостаткам — неприменимость для корней чётной кратности и невозможность обобщения на комплексные числа и системы уравнений. В нашем случае результат, полученный программой, совпал с данным приближением корня вплоть до третьего знака после запятой.

Метод итераций требует найти уравнение, равносильное данному и с выделенным линейным членом. Оно задаёт новый член приближения. Найти такое уравнение, ко всему вышесказанному, ещё и удовлетворяющее условиям сходимости метода, может представиться трудностью — в том недостаток метода итераций.

Метод касательных требует вычисления производной на каждом шаге итерационного процесса, что есть его недостаток вкупе с локальностью: метод гарантированно сходится при произвольном стартовом значении x только тогда, когда $|f(x)\cdot f''(x)|<(f'(x))^2$. Однако сходимость метода Ньютона квадратичная — очень быстрая — , что является его достоинством. В нашем случае результат, полученный программой, совпал с данным приближением корня вплоть до четвёртого, последнего, знака после запятой.

Тем не менее, недостатком почти всех итерационных методов нахождения корней является то, что при однократном применении они позволяют найти лишь один корень функции, к тому же, мы не знаем какой именно. Чтобы найти другие корни, можно было бы брать новые стартовые точки и применять метод вновь, но нет гарантии, что при этом итерации сойдутся к новому корню, а не к уже найденному, если вообще сойдутся.

Список литературы

https://mainfodotru.files.wordpress.com/2017/09/numeric-methods-part2.pdf — численное решение нелинейных уравнений, метод дихотомии

http://www.apmath.spbu.ru/ru/structure/depts/is/task4-2013.pdf, методическая литература к курсовому проекту №4 — описание методов итераций и Ньютона