Национальная научно-образовательная корпорация ИТМО

Факультет программной инженерии и компьютерной техники

ЛАБОРАТОРНАЯ РАБОТА №7

по дисциплине

«Вычислительная математика»

Вариант № 8

Выполнил:

Студент группы P3209

Ляшенко Никита Андреевич

Преподаватель:

Наумова Надежда Александровна

Санкт-Петербург, 2024

## Цель работы

Изучить численные методы решения нелинейных уравнений и их систем, найти корни заданного нелинейного уравнения/системы нелинейных уравнений, выполнить программную реализацию методов.

## Задание

Часть 1.

1. Отделить корни заданного нелинейного уравнения графически (вид уравнения представлен в табл. 6)
2. Определить интервалы изоляции корней.
3. Уточнить корни нелинейного уравнения (см. табл. 6) с точностью *ϵ* = 10−2.
4. Используемые методы для уточнения каждого из 3-х корней многочлена представлены в таблице 7.
5. Вычисления оформить в виде таблиц (1-5), в зависимости от заданного метода. Для всех значений в таблице удержать 3 знака после запятой.
   1. Для метода половинного деления заполнить таблицу 1.
   2. Для метода хорд заполнить таблицу 2.
   3. Для метода Ньютона заполнить таблицу 3.
   4. Для метода секущих заполнить таблицу 4.
   5. Для метода простой итерации заполнить таблицу 5. Проверить условие сходимости метода на выбранном интервале.
6. Заполненные таблицы отобразить в отчете.

Вид нелинейного уравнения для вычислительной реализации:

3*x*3 + 1*,*7*x*2 − 15*,*42*x* + 6*,*89 = 0

Выбор метода для вычислительной реализации задачи:

|  |  |  |  |
| --- | --- | --- | --- |
| Номер | Крайний | Крайний | Центральный |
| варианта | правый корень | левый корень | корень |
| 8 | Метод простой итерации | Метод хорд | Метод Ньютона |

## Выполнение первой части

Точки пересечения:

График функции:

Изображение выглядит как линия, График, диаграмма

Автоматически созданное описание

Метод для правого корня

преобразуем уравнение 𝑓 (𝑥) = 0 к равносильному (при 𝜆 ≠ 0) 𝜆𝑓(𝑥) = 0

𝑓(x)= 3*x*3 + 1*,*7*x*2 − 15*,*42*x* + 6*,*89

x = x + 𝜆𝑓𝑥

ᵩ(x) = 𝑥 + 𝜆𝑓(𝑥),

𝜑′(𝑥) = 1 + 𝜆𝑓′(x)

𝑓′(x) = 9*x*2 + 3*,*4*x* − 15*,*42

*f*′(1*,*5) = 9 · (1*,*5)2 + 3*,*4 · (1*,*5) − 15*,*42 = 9,93

*f*′(2) = 9 · 4 + 3*,*4 · 2 − 15*,*42 = 27*,*38

Так как *f*′[*a,b*] *>* 0, то рассматриваем:

𝜆 = -0,0365

Подставим: *ϕ*(*x*) = *x* -0,0365· (3*x*3 + 1*,*7*x*2 − 15*,*42*x* + 6*,*89) =

*ϕ*′(*x*) =

Проверим точки:

*ϕ*′(1*,*5) = 0.637327*< 1*

*ϕ*′(2) = 0,0006 *<* 1

Сходится

*x*0 = 1*,*5

*x*1 = 1*,*348

*x*2 = 1*,*476

*f*(*x*2) = −2*,*520

...

*f*(*x*) = 3*x*3 + 1*,*7*x*2 − 15*,*42*x* + 6*,*89

*ϕ*(*x*) =

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Номер | *xi* | *xi*+1 | *ϕ*(*xi*+1) | *f*(*xi*+1) | |*xi*+1 − *xi*| |
| 0 | 1,5 | 1,584 | 1,6332 | -1,347 | 0,0584 |
| 1 | 1,584 | 1,6332 | 1,66 | -0,6906 | 0,0492 |
| 2 | 1,6332 | 1,66 | 1,671 | -0.3 | 0,0268 |
| 3 | 1,66 | 1,671 | 1,6758 | -0,1325 | 0,01 |
| 4 | 1,671 | 1,6758 | 1,6787 | -0,058 | 0,0048 |

Таблица 2: Уточнение корня уравнения методом простой итерации

*x* ≈ −1,6787

1. **Метод хорд для левого корня**

Возьму за изолированный интервал [-3, -2.5]

*f*(*x*) = 3*x*3 + 1*,*7*x*2 − 15*,*42*x* + 6*,*89

Вычисление будем производить по формуле:

Изображение выглядит как черный, темнота

Автоматически созданное описание

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Номер | *a* | *b* | *x* | *f*(*a*) | *f*(*b*) | *f*(*x*) | |*xi*+1 − *xi*| |
| 0 | -3 | -2.5 | -2.71136 | -12.55 | 9,19 | 1,39921 |  |
| 1 | -3 | -2.71136 | -2,74031 | -12.55 | 1,39921 | 0,17797 | 0,02895 |
| 2 | -3 | -2,74031 | -2,74394 | -12.55 | 0,17797 | 0,02213 | 0,00363 |
| 3 | -3 | -2,74394 | -2,74439 | -12.55 | 0,02213 |  | 0,00045 |

Таблица 3: Уточнение корня уравнения методом хорд

*x* ≈ −2*.*74439

1. Находим центральный корень

Возьму изолированный интервал [0*.*4*,*0*.*6]

*f*(*x*) = 3*x*3 + 1*.*7*x*2 − 15*.*42*x* + 6*.*89

Найдём производные:

*f*′(*x*) = 9*x*2 + 3*.*4*x* − 15*.*42;

*f*``(*x*) = 18x+ 3*.*4;

*f*(0*.*4) = 1*.*186; *f*(0*.*6) = −1*.*102

*f*′(0*.*4) = −12*.*62; *f*′(0*.*6) = −10*,*14

*f*′′(0*.*4) = 10*.*6; *f*′′(0*.*6) = 14*,*2

Знаки сохраняются.

Выполняется условие *f*(*a*0) · *f*′′(*a*0) *>* 0|=> *x*0 = *a*0 = 0*.*4

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Номер | *xi* | *f*(*xi*) | *f*′(*xi*) | *xi*+1 | |*xi*+1 − *xi*| |
| 0 | 0.4 | 1.186 | -12.62 | 0.49398 | 0.09398 |
| 1 | 0.49398 | 0.049273 | -11.54432 | 0.4982 | 0.00427 |
| 2 | 0.4982 | 0,0000915 | -11.49167 | 0.4982 | 0 |

*x* ≈ 0*.*4982

# Выполнение второй части

Задание:

1. Отделить корни заданной системы нелинейных уравнений графически (вид системы представлен в табл. 8).
2. Используя указанный метод, решить систему нелинейных уравнений с точностью до 0,01.
3. Для метода простой итерации проверить условие сходимости метода.
4. Подробные вычисления привести в отчете.

Система нелинейных уравнений для вычислительной реализации:

*tg x* · *y* = *x*2 0*.*8*x*2 + 2*y*2 = 1

Система имеет не более двух решений, это видно по графику. Решения в точках *x*1*,x*2. Выразим:

Изображение выглядит как линия, График, диаграмма

Автоматически созданное описание

Изображение выглядит как текст, рукописный текст, бумага, документ

Автоматически созданное описание

# Программная реализация задачи

Задачи для нелинейных:

1. Все численные методы (см. табл. 9) должны быть реализованы в виде отдельных подпрограмм/методов/классов.
2. Пользователь выбирает уравнение, корень/корни которого требуется вычислить (3-5 функций, в том числе и трансцендентные), из тех, которые предлагает программа.
3. Предусмотреть ввод исходных данных (границы интервала/начальное приближение к корню и погрешность вычисления) из файла или с клавиатуры по выбору конечного пользователя.
4. Выполнить верификацию исходных данных. Необходимо анализировать наличие корня на введенном интервале.Если на интервале несколько корней или они отсутствуют – выдавать соответствующее сообщение. Программа должна реагировать на некорректные введенные данные.
5. Для методов, требующих начальное приближение к корню (методы Ньютона, секущих, хорд с фиксированнымконцом, простой итерации), выбор начального приближения x0(а или b) вычислять в программе.
6. Для метода простой итерации проверять достаточное условие сходимости метода на введенном интервале.
7. Предусмотреть вывод результатов (найденный корень уравнения, значение функции в корне, число итераций) вфайл или на экран по выбору конечного пользователя.

Мой код: