МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

«НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ

УНИВЕРСИТЕТ им. Р. Е. АЛЕКСЕЕВА»

Кафедра «Вычислительные системы и технологии»

**ВЫЧИСЛИТЕЛЬНАЯ МАТЕМАТИКА**

*Учебно-методическое пособие к лабораторным работам по курсу «Вычислительная математика» для студентов высших учебных заведений   
по направлению 09.03.01 «Информатика и вычислительная техника» профилей «Вычислительные машины, комплексы, системы и сети» и «Программное обеспечение средств вычислительной техники и автоматизированных систем»*

*всех форм обучения*

Нижний Новгород 2022

Составители: **А.С. Суркова, А.З Панкратова**

**УДК 519.6**

**Вычислительная математика:** учебно-метод. пособие к лаб. работам по курсу «Вычислительная математика» для студентов высших учебных заведений по направлению 09.03.01 «Информатика и вычислительная техника» профилей «Вычислительные машины, комплексы, системы и сети» и «Программное обеспечение средств вычислительной техники и автоматизированных систем» всех форм обучения / НГТУ им. Р.Е. Алексеева; сост.: А.С. Суркова, А.З. Панкратова – Нижний Новгород, 2022. – 40 с.

В учебно-методическом пособии описываются лабораторные работы по дисциплине «Вычислительная математика». Каждая лабораторная работа снабжена теоретической частью и подробным описанием хода проведения экспериментов. Изложение сопровождается конкретными примерами выполнения. Материал предназначен для студентов высших учебных заведений по направлению 09.03.01 «Информатика и вычислительная техника» профилей «Вычислительные машины, комплексы, системы и сети» и «Программное обеспечение средств вычислительной техники и автоматизированных систем».

Редактор Э.Б. Абросимова.

Подп. к печ. 03.03.2022. Формат 60х84. Печать офсетная. Бумага газетная. Усл. печ. л. 3.5. Тираж 100 экз. Заказ.

Нижегородский государственный технический университет им. Р.Е. Алексеева.

Типография НГТУ, 603950, Нижний Новгород, ул. Минина, 24.

|  |  |
| --- | --- |
|  | © Нижегородский государственный технический университет  им. Р.Е. Алексеева, 2022 |

Содержание

**1. Лабораторная работа №1. Решение нелинейных уравнений методами деления отрезка пополам, хорд, касательных, простой итерации** 4

1.1. Теоретическая часть…………………………………………………………4

1.2. Задание к лабораторной работе……………………………………………..9

**2.** **Лабораторная работа №2. Решение систем линейных уравнений итерационным методом, методом Гаусса и методом Гаусса-Зейделя.......10**

2.1. Теоретическая часть 10

2.2. Задание к лабораторной работе 15

**3.** **Лабораторная работа №3. Интерполирование функции многочленом**

**Ньютона и многочленом Лагранжа ……** 17

3.1. Теоретическая часть 17

3.2. Задание к лабораторной работе 21

**4.** **Лабораторная работа №4. Численное дифференцирование функций**

**с помощью интерполяционных многочленов Ньютона и Лагранжа……** 26

4.1. Теоретическая часть 26

4.2. Задание к лабораторной работе 28

**5.** **Лабораторная работа №5. Численное интегрирование функций методами прямоугольников, трапеций, Симпсона** 32

5.1. Теоретическая часть 32

5.2. Задание к лабораторной работе 35

**6.** **Лабораторная работа №6. Определение собственных чисел и**

**собственных векторов матрицы методом Крылова** 37

6.1. Теоретическая часть 37

6.2. Задание к лабораторной работе 40

**7.** **Лабораторная работа №7. Приближенное решение обыкновенных дифференциальных уравнений методом Эйлера, модифицированным методом Эйлера и методом Рунге-Кутта…………………………………….**42

7.1. Теоретическая часть 42

7.2. Задание к лабораторной работе 49

**8.** **Лабораторная работа №8. Приближенное решение дифференциальных уравнений методом Адамса…………………………………………………**51

8.1. Теоретическая часть 51

8.2. Задание к лабораторной работе 53

**СПИСОК ЛИТЕРАТУРЫ** 54

**Лабораторная работа № 1. Решение нелинейных уравнений методами деления отрезка пополам, хорд, касательных, простой итерации**

**Цель работы**: изучить численные методы и алгоритмы решения нелинейных уравнений

**Постановка задачи**: реализовать изученные алгоритмы решения нелинейных уравнений и провести сравнение методов.

**1.1. Теоретическая часть.**

Рассмотрим уравнение *f(x)=0*  , где функция ) *f (x)* определена и непрерывна в некотором конечном и бесконечном интервале *a < x < b* .

Корнем уравнения *f(x)=0*  называется значение ξ , обращающее функцию *f (x)* в нуль, т.е. такое, что f (ξ ) = 0 . Уравнение *f(x)=0*  называется алгебраическим, если функция *f (x*) является многочленом

в противном случае уравнение *f (x) = 0* называется трансцендентным.

Чаще всего на практике алгебраические и трансцендентные уравнения не удается решить аналитическими методами. Для решения таких уравнений используется численные методы. Алгоритм нахождения корня уравнений с помощью численного метода состоит из двух этапов:

1. отделение и локализация корня, т.е. установление промежутка *[a,b]* , в котором содержится один корень;

2. уточнение значения корня, т.е. находят корни с заданной точностью.

Теоретическим обоснованием существования корней на промежутке [a,b] являются следующие теоремы.

Теорема 1: Если функция f (x) непрерывна на отрезке [a,b], причем *f (a) ∗ f (b) < 0* , то на этом отрезке существует хотя бы один корень уравнения *f(x)=0*.

Теорема 2: Если непрерывная функция f (x) монотонна на отрезке [a,b] , причем

*f (a) ∗ f (b) < 0* , то на этом отрезке существует единственный корень уравнения *f(x)=0*.

**Отделение и локализация корня. Шаговый метод**

Существует несколько методов отделения корней: геометрический метод, аналитический и шаговый. Рассмотрим шаговый метод локализации корня. Дано уравнение *f(x)=0*. Задан интервал поиска ]. Требуется найти интервал *[a,b]* длиной *h*, содержащий первый корень уравнения, начиная с левой границы интервала поиска.

Алгоритм метода: 1. Установить .

2. Определить координату точки *b* *(b=a+h),* а также значения функции в точках a и b: *f(a)* и *f(b)*.

3. Проверить условие выполнение условия *f (a) ∗ f (b) < 0* . Если условие не выполнено – положить *a=b* и перейти к пункту 2. Если условие выполнено - закончить алгоритм.

После того, как найден интервал локализации корня, применяют итерационные методы уточнения корня. Рассмотрим методы половинного деления, метод Ньютона и метод простой итерации.

**Метод половинного деления**

Метод основан на последовательном сужении интервала, содержащего единственный корень уравнения *f(x)=0*  до тех пор, пока не будет достигнута заданная точность ε. Пусть задан отрезок [a,b], содержащий один корень уравнения. Этот отрезок может быть предварительно найден с помощью шагового метода. Алгоритм метода (рис. 1.1):

1. Определить новое приближение корня x в середине отрезка [a,b]: x=(a+b)/2.

2. Найти значения функции в точках a и x: f(a) и f(x).

3. Проверить выполнение условия *f(a)\*f(x)<0*. Если условие выполнено, то корень расположен на отрезке *[a,x].* В этом случае необходимо точку *b* переместить в точку *x* *(b=x).* Если условие не выполнено, то корень расположен на отрезке *[x,b].* В этом случае необходимо точку a переместить в точку *x (a=x)*.

4. Перейти к пункту 1 и вновь поделить отрезок пополам. Алгоритм продолжить до тех пор, пока не будет выполнено условие *f (x) < ε* .

Y

*x=(a+b)/2*

b

a

X

Рис.1.1 Иллюстрация метода половинного деления

Метод является простым в реализации. Недостаток метода: если на отрезке *[a,b]* содержится более одного корня, то метод не работает, т.к. заранее неизвестно, к какому корню сойдется итерационный процесс.

**Метод Ньютона (метод касательных)**

Задан отрезок [a,b], содержащий корень уравнения *f(x)=0* . Метод Ньютона основан на замене исходной функции  f(x), на каждом шаге поиска, касательной, проведенной к этой функции. Пересечение касательной с осью Ox дает приближение корня. Выберем начальную точку *x0=b* (конец интервала). Находим значение функции в этой точке и проводим к ней касательную, пересечение которой с осью Ox дает нам первое приближение корня  - x1 (Рис.1.2)

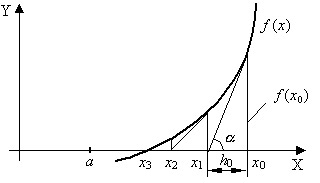


Рис.1.2. Иллюстрация метода Ньютона

, где

Поэтому

В результате итерационный процесс поиска корня описывается рекуррентной формулой

|  |  |
| --- | --- |
|  |  |

Процесс поиска продолжаем до тех пор, пока не выполнится условие: . Как видно из рисунка, метод имеет очень быструю сходимость.

**Метод простой итерации**

Метод основан на замене исходного уравнения *f(x)=0* на эквивалентное *.* Функция *.* выбирается таким образом, чтобы на обоих концах отрезка *[a,b]* выполнялось условие сходимости . В этом случае в качестве начального приближения можно выбрать любой из концов отрезка. Итерационная формула имеет вид

Итерационный процесс продолжается до тех пор, пока не будет выполнено условие .

Пример 1. Методом простых итераций найти приближенное значение корня уравнения с точностью до 0,001 на отрезке [1,2]

Решение. Запишем уравнение в виде:

на отрезке [1,2]. Условие сходимости выполнено, поэтому метод простой итерации применить можно. В качестве начального приближения возьмем Итерационная формула имеет вид:

Для удобства результаты вычислений занесем в таблицу 1.1.

Таблица 1.1. Вычисление корня нелинейного уравнения методом простых итераций

|  |  |  |
| --- | --- | --- |
| *i* | *xi* |  |
| 1 | 2 | -0,30103 |
| 2 | 1,69897 | 0,070844 |
| 3 | 1,769814 | -0,01774 |
| 4 | 1,752072 | 0,004376 |
| 5 | 1,756448 | -0,00108 |
| 6 | 1,755633 | 0,000268 |

Таким образом, искомый корень

**Метод хорд (секущих)**

Метод основан на замене функции f(x)=на каждом шаге поиска хордой, пересечение которой с осью *OX* дает приближение корня.

При этом в процессе поиска семейство хорд может строиться:

а) при фиксированном правом конце хорд, т.е.  начальная точка =a (Рис. 1.3.);

|  |  |  |
| --- | --- | --- |
|  | Итерационная формула: | |
| Рис.3. Иллюстрация метода секущих  а) при фиксированном левом конце хорд, т.е.  начальная точка =b (Рис 1.4.); | | |
|  | | Итерационная формула |

Рис.1.4. Иллюстрация метода секущих

Процесс поиска продолжается до тех пор, пока не выполнится условие

* 1. **Задание к лабораторной работе**

Решить нелинейное уравнение с одним неизвестным с использованием четырех методов (метод половинного деления, метод хорд, метод Ньютона, метод простой итерации). Задание по вариантам. Номер варианта – номер студента в списке группы. Точность ε=0.001

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

***Содержание отчета***

Отчет по данной лабораторной работе должен включать в себя следующие пункты:

1. Тема и цель лабораторной работы;

2. Вариант задания на лабораторную работу;

3. Краткие теоретические сведения и описание алгоритма работы программы в виде блок схемы;

4. Листинг разработанной программы с подробными комментариями;

5. Результаты работы программы;

6. Выводы.

***Контрольные вопросы***

* 1. Понятие численного решения нелинейного уравнения
  2. Методы численного решения нелинейных уравнений
  3. Методы локализации и отделения корней
  4. Отличия методов численного решения нелинейных уравнений

**Лабораторная работа № 2. Решение систем линейных уравнений итерационным методом, методом Гаусса и методом Гаусса-Зейделя**

**Цель работы**: изучить численные методы и алгоритмы решения систем линейных уравнений

**Постановка задачи**: реализовать изученные алгоритмы решения систем линейных уравнений и провести сравнение методов.

**2.1. Теоретическая часть.**

К решению систем линейных уравнений сводятся многочисленные практические задачи. Решение линейных систем является одной из самых распространенных и важных задач вычислительной математики. Значимость задачи породила ряд методов решения, среди этих методов есть как универсальные, так и специализированные, применимые к некоторым системам, имеющим специальные свойства. Однако не существует единого предпочтительного во всех случаях метода.

Рассмотрим систему линейных алгебраических уравнений с неизвестными:

(\*)

Эта система может быть записана в матричном виде: , где

Методы решения систем линейных уравнений делятся на две группы – прямые и итерационные. Прямые методы используют конечные соотношения (формулы), по которым можно найти точное решение систем линейных алгебраических уравнений. Итерационные методы основаны на использовании повторяющегося процесса и позволяют получить решение в результате последовательных приближений.

**Прямые методы решения систем** **линейных уравнений**

*Матричный метод.*

Матричный метод решения применим к системам с ненулевым определителем (). Умножим систему в матричной форме слева на : : .

*Метод Крамера*.

Данный метод также применим только к системам с ненулевым определителем. Если , корни системы уравнений находятся по формуле:

, где .

– определитель матрицы, полученной из матрицы заменой -го столбца столбцом свободных членов.

Указанные методы очень трудоемки для решения систем большой размерности, поскольку нахождение определителей и обратной матрицы требует больших вычислительных мощностей.

*Метод Гаусса*.

Основан на приведении матрицы системы линейных уравнений к треугольному виду (элементы матрицы ниже главной диагонали равны нулю). Треугольный вид матрицы достигается последовательным исключением переменных из уравнений путем элементарных преобразований. Затем находятся все переменные системы последовательно, начиная с последних по номеру.

Более подробно метод Гаусса можно разбить на два этапа:

1 этап – прямой ход. Пусть , тогда с помощью первого уравнения путем элементарных преобразований можно исключить из всех последующих уравнений. Затем, если , с помощью второго уравнения исключаем из всех уравнений, начиная с третьего. Аналогично поступают со всеми остальными уравнениями. Если в процессе исключения переменных на каком-то этапе диагональный элемент оказывается равным нулю (), то необходимо переставить уравнения так, чтобы диагональный элемент был не равен нулю. Если таких уравнений не находится, значит система не является линейно независимой и имеет множество решений.

В результате прямого хода метода Гаусса получается диагональная матрица:

2 этап – обратный ход. Из последнего уравнения выражаем : .

Подставляем полученное выражение в предпоследнее () уравнение и выражаем :

Таким образом последовательно находятся все переменные.

**Итерационные методы решения систем** **линейных уравнений**

*Метод простой итерации (метод Якоби)*.

Приведем систему к виду: , где , . Тогда поиск решения можно представить итерационным процессом:

Рассматриваем начальное приближение , находим приближение на первом шаге, втором и т.д.:

От значения зависит сходимость метода и скорость сходимости алгоритма.

Выразим из каждого -го уравнения системы (\*) переменную :

Возьмем произвольное начальное приближение . Тогда:

Вычисления производят, пока не будет достигнут критерий завершения (остановки). Критериями могут быть условия:

* ;
* .

Справедливо утверждение. Если в системе линейных уравнений матрица А имеет диагональное преобладание, то метод простой итерации сходится при любом начальном приближении. .

*Метод Гаусса-Зейделя*.

Данный метод является модификацией метода простой итерации, основан на использовании информации, получаемой на итерации для расчетов на этой итерации. Формула для расчета переменной на -ой итерации:

Подробнее формулы для вычисления переменных для .

|  |  |
| --- | --- |
|  |  |
| *Метод простой итерации* | *Метод Гаусса-Зейделя* |
|  |  |

**2.2 Задание к лабораторной работе**

Решить систему линейных уравнений методом Гаусса, итерационным методом и методом Гаусса-Зейделя. При необходимости преобразовать систему к диагонально преобладающему виду. Сделать оценку количества итераций для итерационных методов, сравнить. Задание по вариантам. Номер варианта – номер студента в списке группы. Точность ε=0.001.

|  |  |
| --- | --- |
| 1.  3.  5.  7.  9.  11.  13.  15.  17.  19. | 2.  4.  6.  8.  10.  12.  14.  16.  18.  20. |

***Содержание отчета***

Отчет по данной лабораторной работе должен включать в себя следующие пункты:

1. Тема и цель лабораторной работы;

2. Вариант задания на лабораторную работу;

3. Краткие теоретические сведения и описание алгоритма работы программы в виде блок схемы;

4. Листинг разработанной программы с подробными комментариями;

5. Результаты работы программы;

6. Выводы.

***Контрольные вопросы***

* 1. Понятие прямого и итерационного метода решения систем линейных уравнений
  2. Сущность метода Гаусса решения систем линейных уравнений
  3. Сущность метода простой итерации
  4. Сущность метод Гаусса-Зейделя

**Лабораторная работа №3. Интерполирование функции многочленом**

**Ньютона и многочленом Лагранжа**

**Цель работы**: изучить численные методы и алгоритмы решения задачи интерполяции функций

**Постановка задачи**: реализовать изученные алгоритмы интерполяции и провести сравнение методов.

**3.1. Теоретическая часть**

В инженерных расчетах часто требуется установить функцию f(x) для всех значений х отрезка [a,b] , если известны ее значения в некотором конечном числе точек этого отрезка. Одним из способов приближения функции является интерполяция

Задача интерполяции может возникнуть в практике при:

1. интерполировании табличных данных;
2. получении функциональной зависимости по экспериментальным данным, представленным в табличной форме;
3. замене сложной с вычислительной точки зрения функции, более простой зависимостью;
4. при дифференцировании и интегрировании

Пусть на отрезке [x0 ,xn ] заданы n+1 точки x0 , x1 , x2 ,...,xn , называемые узлами интерполяции, и значения некоторой интерполируемой функции y=f(x) в этих точках, т.е. имеется таблица экспериментальных значений функции y=f(x); y0 , y1 , y2 .....yn, где y0=f(x0 ); y1=f(x1 ); ...; yn=f(xn )

Постановка задачи: Требуется найти значения этой функции для промежуточных значений аргумента, не совпадающих с приведенными в таблице. Получить аналитическое выражение функции y= f(x) по таблице ее значений часто бывает невозможно.

Поэтому вместо нее строят другую функцию, которая легко вычисляется и имеет ту же таблицу значений, что и f(x), т.е. Pm(x0 )=f(x0)=y0, …Pm (xi)=f(xi )=yi , где i = 0,1,2, ... , n. Такую задачу называют задачей интерполирования. Точки xi называются узлами интерполяции; функция f(x) называется интерполируемой функцией; многочлен Pm (x) называется интерполяционным многочленом.

**Интерполяционный многочлен Лагранжа для не равностоящих узлов**

Пусть функция *f* задана таблицей 1.

Таблица 1 - Значения функции f(x)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *x* | *xо* | *x\* | *...* | xn |
| *f(х)* | *y0* | *y1* | ... | yn |

Построим интерполяционный многочлен Ln(х), степень которого не больше n и для которого выполнены условия

*F(хо)=у0, F(х1)=у1,* .... *F(хn)=yп.* , (1)

Будем искать *Ln(х)* в виде

*Ln(x)=l0(x)+l1(x)+l2(x)+………+ln(x)* , (2)

где *li(х) —* многочлен степени *п,*

Очевидно, что требование (3) с учетом (2) обеспечивает выполнение условий (1).

Многочлены *li(х)* составим следующим способом:

*li(x)=ci(x-x0)(x-x1)·… ·(x-xi-1)(x-xi+1) ·…·(x-xn),* (4)

где сi - постоянный коэффициент, значение которого можно найти из первой части условия (3). Окончательно получим:

) (5)

Это и есть интерполяционный многочлен Лагранжа для неравностоящих узлов.

Пример 1.

Построить интерполяционный многочлен Лагранжа для функции, заданной таблицей 2

Таблица 2 - Исходные данные

|  |  |  |  |
| --- | --- | --- | --- |
|  | x0 | x1 | x2 |
| x | 1 | 3 | 4 |
| f(x) | 12 | 4 | 6 |

Из таблицы 2 следует, что n=2 (т. е. степень многочлена будет не выше, чем вторая); здесь хо=1, х1=3, х2=4. Используя формулу (5), получаем:

Таким образом, приближающая функция для функции f заданной таблицей 2 имеет следующий вид F(x)=2x2-12x+22.

**Интерполяционный полином Ньютона для равностоящих узлов**

Первая интерполяционная формула Ньютона (интерполирование вперед)

Пусть для функции, заданной таблицей с постоянным шагом**,** составлена таблица конечных разностей таблица 3.

Таблица 3 - Таблица конечных разностей

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| x | y | ∆y1 | ∆2yi | ∆3yi | … |
| x0 | y0 | ∆y0 | ∆2y0 | ∆3y0 |  |
| x1 | y1 | ∆y1 | ∆2y1 | ∆3y1 |  |
| x2 | y2 | ∆y2 | ∆2y2 | … |  |
| x3 | y3 | ∆y3 | … |  |  |
| x4 | y4 | … |  |  |  |
| … | … |  |  |  |  |

Будем искать интерполяционный многочлен в виде:

*Pn(x)=a0 +a1(x-x0)+a2(x-x0)(x-x1) +…+an(x-x0)…(x-xn-1).*(6)

Это многочлен *n*-й степени. Значения коэффициентов *a0, a1, ..., a****n*** найдем из условия совпадения значений исходной функции и многочлена в узлах. Полагая *х=x0*, из (6) находим *yо=Рn(x0)=aо*, откуда *aо=y0*. Далее, придавая *х* значения *х*1,*х*2*,…xn* последовательно получаем:

Подставив в (6), получим:

(7)

Это первая интерполяционная формула Ньютона. Для практического использования интерполяционную формулу Ньютона (7) обычно записывают в несколько преобразованном виде. Для этого введем новую переменную

(8)

Это и есть окончательный вид первой интерполяционной формулы Ньютона. Формулу (8) выгодно использовать для интерполирования функции в окрестности начального значения x0 , где t мало по абсолютной величине, т.е. когда x и x0 мало отличаются.

Если в формуле (8) положить n =1 , то получим формулу линейного интерполирования:

Итак, первая интерполяционная формула Ньютона применяется для интерполирования вблизи левого конца таблицы – для интерполирования вперед и для вычислений x , лежащих вне отрезка за левым концом – экстраполирования назад. Для интерполирования значений вблизи правого конца таблицы применяется вторая интерполяционная формула Ньютона.

**Вторая интерполяционная формула Ньютона**

Для получения второй интерполяционной формулы Ньютона нужно представить искомый многочлен в виде:

(9)

Полагая значения x в формуле (5) равными поочередно: x=xn,…x=x0, можно вычислить все коэффициенты этого многочлена:

Если подставить найденные коэффициенты в формулу (9), получится вторая интерполяционная формула Ньютона. Введем новую переменную t:

(10)

Это и есть обычный вид второй интерполяционной формулы Ньютона. Вторая интерполяционная формула Ньютона применяется при построении многочленов для узлов, расположенных ближе к правому концу таблицы (интерполирование назад) и для узлов, расположенных вне таблицы за правым концом (экстраполирование вперед).

**Задание к лабораторной работе**

Найти приближенное значение функции при данных значениях аргумента с помощью интерполяционного многочлена Лагранжа, если функция задана в неравноотстоящих узлах таблицы, оценить погрешность

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| x | y | № варианта |  |  |
| 0.43 | 1.63597 | 1 | 0.702 | 0.503 |
| 0.48 | 1.73234 | 7 | 0.512 | 0.441 |
| 0.55 | 1.87686 | 13 | 0.645 | 0.602 |
| 0.62 | 2.03345 | 19 | 0.736 | 0.732 |
| 0.70 | 2.22846 |  |  |  |
| 0.75 | 2.35973 |  |  |  |

Таблица 2.2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| x | y | № варианта |  |  |
| 0.02 | 1.02316 | 2 | 0.102 | 0.222 |
| 0.08 | 1.09590 | 8 | 0.114 | 0.092 |
| 0.12 | 1.14725 | 14 | 0.125 | 0.155 |
| 0.17 | 1.21483 | 20 | 0.203 | 0.111 |
| 0.23 | 1.30120 |  |  |  |
| 0.30 | 1.40976 |  |  |  |

Таблица 2.3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| x | y | № варианта |  |  |
| 0.35 | 2.73951 | 3 | 0.526 | 0.444 |
| 0.44 | 2.30080 | 9 | 0.453 | 0.613 |
| 0.47 | 1.96864 | 15 | 0.482 | 0.555 |
| 0.51 | 1.78776 |  |  |  |
| 0.56 | 1.59502 |  |  |  |
| 0.64 | 1.34310 |  |  |  |

Таблица 2.4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| x | y | № варианта |  |  |
| 0.41 | 2.57418 | 4 | 0.616 | 0.444 |
| 0.46 | 2.32513 | 10 | 0.478 | 0.555 |
| 0.52 | 2.09336 | 16 | 0.665 | 0.714 |
| 0.60 | 1.86208 |  |  |  |
| 0.65 | 1.74926 |  |  |  |
| 0.72 | 1.62098 |  |  |  |

Таблица 2.5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| x | y | № варианта |  |  |
| 0.68 | 0.80866 | 5 | 0.896 | 0.603 |
| 0.73 | 0.89492 | 11 | 0.812 | 0.777 |
| 0.80 | 1.02964 | 17 | 0.774 | 0.906 |
| 0.88 | 1.20966 |  |  |  |
| 0.93 | 1.34087 |  |  |  |
| 0.99 | 1.52368 |  |  |  |

Таблица 2.6

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| x | y | № варианта |  |  |
| 0.11 | 9.05421 | 6 | 0.314 | 0.222 |
| 0.15 | 6.61659 | 12 | 0.235 | 0.377 |
| 0.21 | 4.69170 | 18 | 0.332 | 0.123 |
| 0.29 | 3.35106 |  |  |  |
| 0.35 | 2.73951 |  |  |  |
| 0.40 | 2.36522 |  |  |  |

Используя первую или вторую интерполяционную формулу Ньютона, вычислить значения функции при данных значениях аргумента

Таблица 1

|  |  |
| --- | --- |
| x | y |
| 1.415 | 0.888551 |
| 1.420 | 0.889599 |
| 1.425 | 0.890637 |
| 1.430 | 0.891667 |
| 1.435 | 0.892687 |
| 1.440 | 0.893698 |
| 1.445 | 0.894700 |
| 1.450 | 0.895693 |
| 1.455 | 0.896677 |
| 1.460 | 0.897653 |
| 1.465 | 0.898619 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № вари-  анта | Значение аргумента | | | |
| x1 | x2 | x3 | x4 |
| 1 | 1.4161 | 1.4625 | 1.4135 | 1.470 |
| 11 | 1.4179 | 1.4633 | 1.4124 | 1.4655 |
| 21 | 1.4263 | 1.4575 | 1.410 | 1.4662 |

Таблица 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № вари-  анта | Значение аргумента | | | |
| x1 | x2 | x3 | x4 |
| 2 | 0.1026 | 0.1440 | 0.099 | 0.161 |
| 12 | 0.1035 | 0.1492 | 0.096 | 0.153 |
| 22 | 0.1074 | 0.1485 | 0.1006 | 0.156 |

|  |  |
| --- | --- |
| x | y |
| 0.101 | 1.26183 |
| 0.106 | 1.27644 |
| 0.111 | 1.29122 |
| 0.116 | 1.306617 |
| 0.121 | 1.32130 |
| 0.126 | 1.33660 |
| 0.131 | 1.35207 |
| 0.136 | 1.36773 |
| 0.141 | 1.38357 |
| 0.146 | 1.39959 |
| 0.151 | 1.41579 |

Таблица 3

|  |  |
| --- | --- |
| x | y |
| 0.15 | 0.860708 |
| 0.20 | 0.818731 |
| 0.25 | 0.778801 |
| 0.30 | 0.740818 |
| 0.35 | 0.704688 |
| 0.40 | 0.670320 |
| 0.45 | 0.637628 |
| 0.50 | 0.606531 |
| 0.55 | 0.576950 |
| 0.60 | 0.548812 |
| 0.65 | 0.522046 |
| 0.70 | 0.496585 |
| 0.75 | 0.4722367 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № вари-  анта | Значение аргумента | | | |
| x1 | x2 | x3 | x4 |
| 3 | 0.1511 | 0.7250 | 0.1430 | 0.80 |
| 13 | 0.1535 | 0.7333 | 0.100 | 0.7540 |
| 23 | 0.1525 | 0.6730 | 0.1455 | 0.85 |

Таблица 4

|  |  |
| --- | --- |
| x | y |
| 0.180 | 5.61543 |
| 0.185 | 5.46693 |
| 0.190 | 5.32634 |
| 0.195 | 5.19304 |
| 0.200 | 5.06649 |
| 0.205 | 4.94619 |
| 0.210 | 4.83170 |
| 0.215 | 4.72261 |
| 0.220 | 4.61855 |
| 0.225 | 4.51919 |
| 0.230 | 4.42422 |
| 0.235 | 4.33337 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № вари-  анта | Значение аргумента | | | |
| x1 | x2 | x3 | x4 |
| 4 | 0.1817 | 0.2275 | 0.175 | 0.2375 |
| 14 | 0.1827 | 0.2292 | 0.1776 | 0.240 |
| 24 | 0.1873 | 0.2326 | 0.1783 | 0.245 |

Таблица 5

|  |  |
| --- | --- |
| x | y |
| 3.50 | 33.1154 |
| 3.55 | 34.8133 |
| 3.60 | 36.5982 |
| 3.65 | 38.4747 |
| 3.70 | 40.4473 |
| 3.75 | 42.5211 |
| 3.80 | 44.7012 |
| 3.85 | 46.9931 |
| 3.90 | 49.4024 |
| 3.95 | 51.5982 |
| 4.00 | 57.3975 |
| 4.10 | 60.3403 |
| 4.15 | 63.4340 |
| 4.20 | 66.6863 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № вари-  анта | Значение аргумента | | | |
| x1 | x2 | x3 | x4 |
| 5 | 3.522 | 4.176 | 3.475 | 4.25 |
| 15 | 3.573 | 4.184 | 3.488 | 4.30 |
| 25 | 3.575 | 4.142 | 3.45 | 4.204 |

Таблица 6

|  |  |
| --- | --- |
| x | y |
| 0.115 | 8.65729 |
| 0.120 | 8.29329 |
| 0.125 | 7.95829 |
| 0.130 | 7.64893 |
| 0.135 | 7.36235 |
| 0.140 | 7.09613 |
| 0.145 | 6.84815 |
| 0.150 | 6.61659 |
| 0.155 | 6.39986 |
| 0.160 | 6.19658 |
| 0.165 | 6.00551 |
| 0.170 | 5.82558 |
| 0.175 | 5.65583 |
| 0.180 | 5.49543 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № вари- анта | Значение аргумента | | | |
| x1 | x2 | x3 | x4 |
| 6 | 0.1217 | 0.1736 | 0.1141 | 0.185 |
| 16 | 0.1168 | 0.1745 | 0.110 | 0.1825 |
| 26 | 0.1175 | 0.1773 | 0.1134 | 0.190 |

Таблица 7

|  |  |
| --- | --- |
| x | y |
| 1.340 | 4.25562 |
| 1.345 | 4.35325 |
| 1.350 | 4.45522 |
| 1.355 | 4.56184 |
| 1.360 | 4.67344 |
| 1.365 | 4.79038 |
| 1.370 | 4.91306 |
| 1.375 | 5.04192 |
| 1.380 | 5.17744 |
| 1.385 | 5.32016 |
| 1.390 | 5.47069 |
| 1.395 | 5.62968 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № вари-  анта | Значение аргумента | | | |
| x1 | x2 | x3 | x4 |
| 7 | 1.3617 | 1.3921 | 1.3359 | 1.400 |
| 17 | 1.3463 | 1.3868 | 1.335 | 1.3990 |
| 27 | 1.3432 | 1.3936 | 1.3365 | 1.3975 |

Таблица 8

Таблица 9

|  |  |
| --- | --- |
| x | y |
| 0.01 | 0.991824 |
| 0.06 | 0.951935 |
| 0.11 | 0.913650 |
| 0.16 | 0.876905 |
| 0.21 | 0.841638 |
| 0.26 | 0.807789 |
| 0.31 | 0.775301 |
| 0.36 | 0.744120 |
| 0.41 | 0.714193 |
| 0.46 | 0.685470 |
| 0.51 | 0.657902 |
| 0.56 | 0.631442 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № вари-  анта | Значение аргумента | | | |
| x1 | x2 | x3 | x4 |
| 8 | 0.027 | 0.525 | 0.008 | 0.61 |
| 18 | 0.1243 | 0.462 | 0.0094 | 0.66 |
| 28 | 0.083 | 0.5454 | 0.0075 | 0.573 |

|  |  |
| --- | --- |
| x | y |
| 0.15 | 4.4817 |
| 0.16 | 4.9530 |
| 0.17 | 5.4739 |
| 0.18 | 6.0496 |
| 0.19 | 6.6859 |
| 0.20 | 7.3891 |
| 0.21 | 8.1662 |
| 0.22 | 9.0250 |
| 0.23 | 9.9742 |
| 0.24 | 11.0232 |
| 0.25 | 12.1825 |
| 0.26 | 13.4637 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № вари-  анта | Значение аргумента | | | |
| x1 | x2 | x3 | x4 |
| 9 | 0.1539 | 0.2569 | 0.14 | 0.2665 |
| 19 | 0.1732 | 0.2444 | 0.1415 | 0.27 |
| 29 | 0.1648 | 0.2550 | 0.1387 | 0.28 |

Таблица 10

|  |  |
| --- | --- |
| x | y |
| 0.45 | 20.1946 |
| 0.46 | 19.6133 |
| 0.47 | 18.9425 |
| 0.48 | 18.1746 |
| 0.49 | 17.3010 |
| 0.50 | 16.3123 |
| 0.51 | 15.1984 |
| 0.52 | 13.9484 |
| 0.53 | 12.5508 |
| 0.54 | 10.9937 |
| 0.55 | 9.2647 |
| 0.56 | 7.3510 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № вари-  анта | Значение аргумента | | | |
| x1 | x2 | x3 | x4 |
| 10 | 0.455 | 0.5575 | 0.44 | 0.5674 |
| 20 | 0.4732 | 0.5568 | 0.445 | 0.57 |
| 30 | 0.4675 | 0.5511 | 0.4423 | 0.58 |

***Содержание отчета***

1. Тема и цель лабораторной работы;

2. Вариант задания на лабораторную работу;

3. Краткие теоретические сведения и описание алгоритма работы программы в виде блок схемы;

4. Листинг разработанной программы с подробными комментариями;

5. Результаты работы программы;

6. Выводы.

**Контрольные вопросы**

1. Что такое интерполяция?
2. Сформулируйте задачу интерполяции.
3. Интерполяционный многочлен Лагранжа для не равностоящих узлов.
4. Что называется разностью первого порядка, второго порядка и т. д.
5. Первая интерполяционная формула Ньютона(вперед).
6. Вторая интерполяционная формула Ньютона(назад).

**Лабораторная работа № 4. Численное дифференцирование функций**

**с помощью интерполяционных многочленов Ньютона и Лагранжа**

**Цель работы *з***акрепление знаний и умений по численному дифференцированию функций с помощью интерполяционного многочлена Ньютона и метода неопределенных коэффициентов.

**Постановка задачи**: реализовать изученные алгоритмы интерполяции и провести сравнение методов.

**4.1. Теоретическая часть.**

Численное дифференцирование применяется в тех случаях, когда: функция f(x) задана таблично и, следовательно, методы дифференциального исчисления неприменимы; аналитическое выражение f(x) столь сложно, что вычисления производной представляют значительны трудности. В основе численного дифференцирования лежит следующий прием: исходная функция f(x) заменяется на рассматриваемом отрезке [a,b] интерполяционным полиномом Pn(x) и считается, что f’(x) и P’n(x) примерно равны, т.е. f’(x)=P’n(x).

Всегда, когда это возможно, для численного дифференцирования используется интерполяционный многочлен с равноотстоящими узлами, так как это значительно упрощает формулы численного дифференцирования. При равноотстоящих узлах строится интерполяционный полином Ньютона или Лагранжа, а затем он дифференцируется.

При численном дифференцировании интерполяционный полином строится не по всем узлам таблицы, а по трем-пяти узлам, близлежащим к точке, в которой требуется вычислить производную. Если требуется вычислить производную во всех узлах, то вначале полином строится по первым 3-5 узлам и в них вычисляется производная, потом полином строится по следующим 3-5 узлам и в них вычисляется производная и т. д. до тех пор, пока не будет просчитана вся таблица.

**Численное дифференцирование, основанное на многочлене Лагранжа для равноотстоящих узлов:**

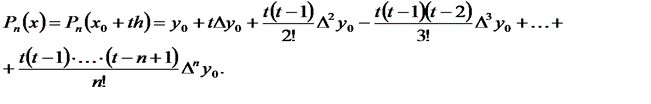
Для квадратичной интерполяции (n=2, i=(0,1,2)):

Вычислим значения производных в узлах, при n = 2 :

и так далее.

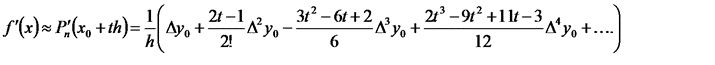
**Численное дифференцирование, основанное на многочлене Ньютона для равноотстоящих узлов:**

Запишем для функции f(x), заданной своими значениями в равноотстоящих узлах https://pdnr.ru/infopediasu/baza25/4265425561280.files/image358.gif первый интерполяционный многочлен Ньютона:

Перепишем этот полином, производя перемножение скобок:

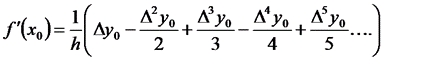
https://pdnr.ru/infopediasu/baza25/4265425561280.files/image362.gif

Дифференцируя https://pdnr.ru/infopediasu/baza25/4265425561280.files/image364.gif по t , получим:



Подобным путем можно получить и производные функции f(x) более высоких порядков. Однако каждый раз, вычисляя значение производной функции f(x) в фиксированной точке х, в качестве х0 следует брать ближайшее слева узловое значение аргумента.

Если исходным значением х оказывается один из узлов таблицы, то в этом случае каждый узел можно считать начальным, тогда, принимая х=х0, t=0, получаем:



**Задание к лабораторной работе**

Найти первую и вторую производную функции в точках х, заданных таблицей, используя интерполяционные многочлены Ньютона. Сравнить со значениями производных, вычисленными по формулам, основанным на интерполировании многочленом Лагранжа.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1.   |  |  | | --- | --- | | x | y | | 1.415 | 0.88551 | | 1.420 | 0.89599 | | 1.425 | 0.90637 | | 1.430 | 0.91667 | | 1.435 | 0.92687 | | 1.440 | 0.93698 | | 1.445 | 0.94700 | | 1.450 | 0.96677 | | 1.455 | 0.97653 | | 1.460 | 0.97653 | | 1.465 | 0.98619 | | 2.   |  |  | | --- | --- | | x | y | | 0.101 | 1.26183 | | 0.106 | 1.27644 | | 0.111 | 1.29122 | | 0.116 | 1.30617 | | 0.121 | 1.32130 | | 0.126 | 1.33660 | | 0.131 | 1.35207 | | 0.136 | 1.36773 | | 0.141 | 1.38357 | | 0.146 | 1.39959 | | 0.151 | 1.41579 | |
| 3.   |  |  | | --- | --- | | x | y | | 0.15 | 0.860708 | | 0.20 | 0.818731 | | 0.25 | 0.778801 | | 0.30 | 0.740818 | | 0.35 | 0.704688 | | 0.40 | 0.670320 | | 0.45 | 0.606531 | | 0.50 | 0.576950 | | 0.55 | 0.548812 | | 0.60 | 0.522046 | | 0.65 | 0.496585 | | 0.70 | 0.496585 | | 0.75 | 0.472367 | | 4.   |  |  | | --- | --- | | x | y | | 0.180 | 5.61543 | | 0.185 | 5.46693 | | 0.190 | 5.32634 | | 0.195 | 5.19304 | | 0.200 | 5.06649 | | 0.205 | 4.94619 | | 0.210 | 4.83170 | | 0.215 | 4.72261 | | 0.220 | 4.61855 | | 0.230 | 4.42422 | | 0.235 | 4.33337 | |
| 5.   |  |  | | --- | --- | | x | y | | 3.50 | 33.1154 | | 3.55 | 34.8133 | | 3.60 | 36.5982 | | 3.65 | 38.4747 | | 3.70 | 40.4473 | | 3.75 | 42.5211 | | 3.80 | 44.7012 | | 3.85 | 46.9931 | | 3.90 | 49.4024 | | 3.95 | 51.9354 | | 4.00 | 54.5982 | | 4.05 | 57.3975 | | 4.10 | 60.3403 | | 4.15 | 63.4340 | | 4.20 | 66.6863 | | 6.   |  |  | | --- | --- | | x | y | | 0.115 | 8.65729 | | 0.120 | 8.29329 | | 0.125 | 7.95829 | | 0.130 | 7.64893 | | 0.135 | 7.36235 | | 0.140 | 7.09613 | | 0.145 | 6.84815 | | 0.150 | 6.61659 | | 0.155 | 6.39386 | | 0.160 | 6.19658 | | 0.165 | 6.00551 | | 0.170 | 5.82558 | | 0.175 | 5.65583 | | 0.180 | 5.49543 | |
| 7.   |  |  | | --- | --- | | x | y | | 1.340 | 4.25562 | | 1.345 | 4.35325 | | 1.350 | 4.45522 | | 1.355 | 4.56184 | | 1.360 | 4.67344 | | 1.365 | 4.79038 | | 1.370 | 4.91306 | | 1.375 | 5.04192 | | 1.380 | 5.17744 | | 1.385 | 5.32016 | | 1.390 | 5.47069 | | 1.395 | 5.62968 | | 8.   |  |  | | --- | --- | | x | y | | 0.01 | 0.991824 | | 0.06 | 0.951935 | | 0.11 | 0.913650 | | 0.16 | 0.876905 | | 0.21 | 0.841638 | | 0.26 | 0.807789 | | 0.31 | 0.775301 | | 0.36 | 0.744120 | | 0.41 | 0.714193 | | 0.46 | 0.685470 | | 0.51 | 0.657902 | | 0.56 | 0.631442 | |
| 9.   |  |  | | --- | --- | | x | y | | 0.15 | 4.4817 | | 0.16 | 4.9530 | | 0.17 | 5.4739 | | 0.18 | 6.0496 | | 0.19 | 6.6859 | | 0.20 | 7.3891 | | 0.21 | 8.1662 | | 0.22 | 9.0250 | | 0.23 | 9.9742 | | 0.24 | 11.0232 | | 0.25 | 12.1825 | | 0.26 | 13.4637 | | 10.   |  |  | | --- | --- | | x | y | | 0.45 | 20.1946 | | 0.46 | 19.6133 | | 0.47 | 18.9425 | | 0.48 | 18.1746 | | 0.49 | 17.3010 | | 0.50 | 16.3123 | | 0.51 | 15.1984 | | 0.52 | 13.9484 | | 0.53 | 12.5508 | | 0.54 | 10.9937 | | 0.55 | 9.2647 | | 0.56 | 7.3510 | |
| 11.   |  |  | | --- | --- | | x | y | | 1.415 | 0.88551 | | 1.420 | 0.89599 | | 1.425 | 0.90637 | | 1.430 | 0.91667 | | 1.435 | 0.92687 | | 1.440 | 0.93698 | | 1.445 | 0.94700 | | 1.450 | 0.96677 | | 1.455 | 0.97653 | | 1.460 | 0.97653 | | 1.465 | 0.98619 | | 12.   |  |  | | --- | --- | | x | y | | 0.101 | 1.26183 | | 0.106 | 1.27644 | | 0.111 | 1.29122 | | 0.116 | 1.30617 | | 0.121 | 1.32130 | | 0.126 | 1.33660 | | 0.131 | 1.35207 | | 0.136 | 1.36773 | | 0.141 | 1.38357 | | 0.146 | 1.39959 | | 0.151 | 1.41579 | |
| 13.   |  |  | | --- | --- | | x | y | | 0.15 | 0.860708 | | 0.20 | 0.818731 | | 0.25 | 0.778801 | | 0.30 | 0.740818 | | 0.35 | 0.704688 | | 0.40 | 0.670320 | | 0.45 | 0.606531 | | 0.50 | 0.576950 | | 0.55 | 0.548812 | | 0.60 | 0.522046 | | 0.65 | 0.496585 | | 0.70 | 0.472367 | | 0.75 | 0.447937 | | 14.   |  |  | | --- | --- | | x | y | | 0.180 | 5.61543 | | 0.185 | 5.46693 | | 0.190 | 5.32634 | | 0.195 | 5.19304 | | 0.200 | 5.06649 | | 0.205 | 4.94619 | | 0.210 | 4.83170 | | 0.215 | 4.72261 | | 0.220 | 4.61855 | | 0.230 | 4.42422 | | 0.235 | 4.33337 | |
| 15.   |  |  | | --- | --- | | x | y | | 3.50 | 33.1154 | | 3.55 | 34.8133 | | 3.60 | 36.5982 | | 3.65 | 38.4747 | | 3.70 | 40.4473 | | 3.75 | 42.5211 | | 3.80 | 44.7012 | | 3.85 | 46.9931 | | 3.90 | 49.4024 | | 3.95 | 51.9354 | | 4.00 | 54.5982 | | 4.05 | 57.3975 | | 4.10 | 60.3403 | | 4.15 | 63.4340 | | 4.20 | 66.6863 | | 16.   |  |  | | --- | --- | | x | y | | 0.115 | 8.65729 | | 0.120 | 8.29329 | | 0.125 | 7.95829 | | 0.130 | 7.64893 | | 0.135 | 7.36235 | | 0.140 | 7.09613 | | 0.145 | 6.84815 | | 0.150 | 6.61659 | | 0.155 | 6.39386 | | 0.160 | 6.19658 | | 0.165 | 6.00551 | | 0.170 | 5.82558 | | 0.175 | 5.65583 | | 0.180 | 5.49543 | |
| 17.   |  |  | | --- | --- | | x | y | | 1.340 | 4.25562 | | 1.345 | 4.35325 | | 1.350 | 4.45522 | | 1.355 | 4.56184 | | 1.360 | 4.67344 | | 1.365 | 4.79038 | | 1.370 | 4.91306 | | 1.375 | 5.04192 | | 1.380 | 5.17744 | | 1.385 | 5.32016 | | 1.390 | 5.47069 | | 1.395 | 5.62968 | | 18.   |  |  | | --- | --- | | x | y | | 0.01 | 0.991824 | | 0.06 | 0.951935 | | 0.11 | 0.913650 | | 0.16 | 0.876905 | | 0.21 | 0.841638 | | 0.26 | 0.807789 | | 0.31 | 0.775301 | | 0.36 | 0.744120 | | 0.41 | 0.714193 | | 0.46 | 0.685470 | | 0.51 | 0.657902 | | 0.56 | 0.631442 | |
| 19. Таблица 9   |  |  | | --- | --- | | x | y | | 0.15 | 4.4817 | | 0.16 | 4.9530 | | 0.17 | 5.4739 | | 0.18 | 6.0496 | | 0.19 | 6.6859 | | 0.20 | 7.3891 | | 0.21 | 8.1662 | | 0.22 | 9.0250 | | 0.23 | 9.9742 | | 0.24 | 11.0232 | | 0.25 | 12.1825 | | 0.26 | 13.4637 | | 20.   |  |  | | --- | --- | | x | y | | 0.45 | 20.1946 | | 0.46 | 19.6133 | | 0.47 | 18.9425 | | 0.48 | 18.1746 | | 0.49 | 17.3010 | | 0.50 | 16.3123 | | 0.51 | 15.1984 | | 0.52 | 13.9484 | | 0.53 | 12.5508 | | 0.54 | 10.9937 | | 0.55 | 9.2647 | | 0.56 | 7.3510 | |

***Содержание отчета***

1. Тема и цель лабораторной работы;

2. Вариант задания на лабораторную работу;

3. Краткие теоретические сведения и описание алгоритма работы программы в виде блок схемы;

4. Листинг разработанной программы с подробными комментариями;

5. Результаты работы программы;

6. Выводы.

***Контрольные вопросы***

* 1. Понятие численного дифференцирования
  2. Интерполяционные полиномы Ньютона и Лагранжа
  3. Приближенные формулы для вычисления первой и второй производной функции

**Лабораторная работа № 5. Численное интегрирование функций методами прямоугольников, трапеций, Симпсона**

**Цель работы**: изучить численные методы и алгоритмы численного интегрирования

**Постановка задачи**: реализовать изученные алгоритмы численного интегрирования и провести сравнение методов.

**5.1. Теоретическая часть**

К численному интегрированию обращаются, когда требуется вычислить определённый интеграл от функций, заданных таблично, или функций, непосредственное нахождение первообразной которых затруднительно (интеграл не берётся в элементарных функциях). В этом случае значение интеграла можно найти только приближенно, используя тот или иной способ численного интегрирования.

Численное интегрирование основано на замене подынтегральной функции суммой вида: . Такая замена следует из определения интеграла как предела суммы . Приближенное равенство   называется квадратурной формулой,  - узлы квадратурной формулы,  - коэффициенты квадратурной формулы. В зависимости от способа интерполяции подынтегральной функции различают разные методы численного интегрирования (методы прямоугольников, трапеций, парабол и др.). В качестве точки может выбираться любая точка в интервале . В зависимости от выбора этой точки различают методы левых, правых и центральных прямоугольников. Если – левая граница интервала, получаем метод левых прямоугольников, если – середина интервала, метод средних прямоугольников и т.д.

**Методы численного интегрирования**

Вычислим значение определенного интеграла   для заданной на отрезке [*a*, *b*] функции *f*(*x*).

Разобьем интервал [a, b] на *n* отрезков. Обозначая , - длина i-го отрезка,, получаем формулы численного интегрирования.

Метод **левых** прямоугольников: .

Метод **правых** прямоугольников: .

Метод **центральных (средних)** прямоугольников:

, .

Метод **трапеций**:

.

При интегрировании с постоянным шагом :

|  |  |
| --- | --- |
| Метод **левых прямоугольников** |  |
| Метод **правых** прямоугольников |  |
| Метод **центральных (средних) прямоугольников** |  |
| Метод **трапеций** |  |

Метод **Симпсона** (*n* – четное, ):

**Погрешность формул численного интегрирования**

Формула левых и правых прямоугольников

Формула средних прямоугольников

Формула Симпсона

Пример 1.

Вычислить интеграл: , *n*=10

Аналитически: 

Составим таблицу для вычислений

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| i |  |  |  |  |
| 0 | 0 | 1.000 |  |  |
| 1 | 0.1 | 0.990 | 0.05 | 0.998 |
| 2 | 0.2 | 0.962 | 0.15 | 0.978 |
| 3 | 0.3 | 0.917 | 0.25 | 0.941 |
| 4 | 0.4 | 0.862 | 0.35 | 0.891 |
| 5 | 0.5 | 0.800 | 0.45 | 0.832 |
| 6 | 0.6 | 0.735 | 0.55 | 0.768 |
| 7 | 0.7 | 0.671 | 0.65 | 0.703 |
| 8 | 0.8 | 0.610 | 0.75 | 0.640 |
| 9 | 0.9 | 0.552 | 0.85 | 0.581 |
| 10 | 1 | 0.500 | 0.95 | 0.526 |

Применим формулу **левых** прямоугольников:

I=0.1(1.000 + 0.990 + 0.962 + 0.917 + 0.862 + 0.800+ 0.735 + 0.671 + 0.610 + 0.552) = **0.81**

По формуле **правых** прямоугольников:

I=0.1(0.990 + 0.962 + 0.917 + 0.862 + 0.800+ 0.735 + 0.671 + 0.610 + 0.552 + 0.500) = **0.76**

Применим формуле **средних** прямоугольников:

I=0.1(0.998 + 0.978 + 0.941 + 0.891 + 0.832 + 0.768 + 0.703 + 0.640 + 0.581 + 0.526) = **0.7856**

Применим формулу **трапеций**:

I=0.1([(1.0 + 0.5)/2]+ 0.990 + 0.962 + 0.917 + 0.862 + 0.800+ 0.735 + 0.671 + 0.610 + 0.552) = **0.785**

Применим формулу **Симпсона**:

I= [1.000 + 0.500 + 4(0.990 + 0.917 + 0.800+ 0.671 + 0.552) + 2(0.962 + 0.862 + 0.735 + 0.610)] = = **0.7854**

**5.2. Задание к лабораторной работе**

Вычислить интеграл по формулам центральных (средних) прямоугольников, трапеций и формуле Симпсона, при n=8 и n=20; оценить погрешность результата.

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

***Содержание отчета***

1. Тема и цель лабораторной работы;
2. Вариант задания на лабораторную работу;
3. Краткие теоретические сведения и описание алгоритма работы программы в виде блок схемы;
4. Листинг разработанной программы с подробными комментариями;
5. Результаты работы программы;
6. Выводы.

***Контрольные вопросы***

1. Понятие численного интегрирования
2. Методы численного интегрирования
3. Отличия методов численного интегрирования
4. Методы левых, правых и средних прямоугольников
5. Метод трапеций
6. Метод Симпсона, точность метода Симпсона

**Лабораторная работа № 6**

**«Определение собственных чисел и собственных векторов матрицы методом Крылова»**

**Цель работы**: изучить численные методы и алгоритмы определения собственных чисел и собственных векторов матрицы

**Постановка задачи**: реализовать изученные алгоритмы и провести сравнение методов.

**6.1. Теоретическая часть**

Большое число научно-технических задач, а также исследования в области вычислительной математики требуют нахождения собственных значений и собственных векторов матриц.

Рассмотрим квадратную матрицу размерности с действительными элементами.

Матрица называется характеристическая матрица матрицы .

Ее определитель – характеристический определитель – является многочленом -й степени – характеристическим многочленом матрицы . При этом коэффициент при старшей степени .

Характеристический многочлен:

.

Корни характеристического многочлена () – (действительные или мнимые) называются характеристическими или собственными числами или собственными значениями матрицы .

Ненулевой вектор называется собственным вектором матрицы , если . Каждому собственному значению соответствует свой вектор .

Для определения собственных векторов рассматривают характеристическое уравнение: .

Характеристическое уравнение записано в векторной форме, ему соответствует система уравнений:

Определитель этой системы равен нулю (по определению собственных значений ), поэтому система имеет множество решений.

Подставляя в систему каждое конкретное собственное значение и решая систему, находим собственный вектор , соответствующий .

Существует несколько способов нахождения собственных значений и собственных векторов.

***1. Метод непосредственного развертывания***

1) Пусть . Рассматриваем матрицу .

Рассмотрим характеристическую матрицу: . Найдем ее определитель, чтобы найти характеристический многочлен:

Заметим, что коэффициент при равен сумме диагональных элементов матрицы , а свободный член многочлена – определитель матрицы .

2) Пусть . Рассматриваем матрицу , найдем характеристический многочлен ():

сумма диагональных элементов матрицы ,

сумма диагональных миноров второго порядка,

– определитель матрицы .

В общем случае.

сумма диагональных элементов матрицы ,

сумма диагональных миноров второго порядка,

сумма диагональных миноров третьего порядка,

…

определитель матрицы .

Однако метод непосредственного развертывания становится очень трудоемким и ресурсоемким с увеличением размерности матрицы. Поэтому предложены и другие методы.

***2. Метод Крылова***

Метод Крылова для нахождения собственных значений и собственных векторов матрицы основан на теореме Гамильтона-Кэли, согласно которой любая квадратная матрица является корнем своего характеристического многочлена. Если характеристический многочлен матрицы , то .

Рассмотрим характеристический многочлен матрицы в виде:

. Подставляя в него матрицу , получаем уравнение:

(1)

Рассмотрим произвольный ненулевой вектор , и умножим на него справа уравнение (1):

(2)

Положим: , тогда: , , … . Подставим эти выражения в (2):

или

.

Рассматривая покомпонентную запись вектора получаем систему относительно коэффициентов () характеристического многочлена:

(3)

Элементы вычисляются по формулам:

; ; …..; .

Координаты первоначального вектора выбираются произвольно, например: . Если полученная система (3) не имеет единственного решения, то необходимо выбрать другой начальный вектор .

После нахождения коэффициентов характеристического многочлена, могут быть найдены его корни – собственные значения .

Собственные векторы находятся по формуле:

, .

Коэффициенты (,) определяются по схеме Горнера по формулам:

,

***3. Метод Данилевского***

Метод Данилевского для нахождения собственных значений и собственных векторов матрицы основан на преобразованиях подобия матриц.

Матрицы и называются подобными (), если одна получается из другой путем умножения с помощью неособенной матрицы : . Неособенная или невырожденная матрица – квадратная матрица, определитель которой не равен нулю. Подобные матрицы имеют одинаковые характеристические многочлены (соответственно, и одинаковые собственные значения).

В методе Данилевского матрица преобразуется к подобной матрице вида (нормальная форма Фробениуса) :

.

Разложим определитель по первой строке: . Таким образом в матрице первая строка содержит являющиеся обратными коэффициентами характеристического многочлена , а все остальные элементы равны нулю, кроме элементов под главной диагональю, которые равны 1.

Схема метода Данилевского для преобразования матрицы к нормальной форме Фробениуса. Основные операции

1) Рассматриваем матрицу :

Ее последнюю строку необходимо преобразовать к виду: .

Предположим, что элемент , разделим на этот элемент все элементы ()-го столбца матрицы .

2) Вычтем из всех остальных столбцов матрицы ()-й столбец, умноженный соответственно на числа .

3) В качестве неособенной матрицы рассматриваем матрицу , которая получается из единичной матрицы путем таких же преобразований:

|  |  |
| --- | --- |
| , | где  (); |

Матрица получена умножением матрицы на матрицу справа:

|  |  |  |
| --- | --- | --- |
| , | | , при  , при |
| 4) Матрица не является подобной матрице . Умножим матрицу слева на – обратную матрицу к матрице : . Матрица имеет вид: |  | |

|  |  |
| --- | --- |
| При этом матрица   является подобной матрице и имеет последнюю строку, приведенную к форме Фробениуса: |  |

Если , то повторяем операции 1) – 4), взяв за основу ()-ю строчку матрицы : , . Операции повторяем до тех пор, пока не придем к форме Фробениуса .

Как уже было сказано характеристический многочлен матрицы совпадает с характеристическим многочленом исходной матрицы , соответственно равны и их собственные значения .

Рассмотрим конкретное собственное значение и найдем соответствующий ему – собственный вектор матрицы . По определению собственных векторов .

Умножая матрицы можно получить систему уравнений:

Положим , тогда , , …, т.е.

Для нахождения собственного вектора матрицы , соответствующего собственному значению необходимо вектор умножить слева на матрицы , , , которые были рассчитаны при преобразованиях по методу Данилевского.

.

Особенные случаи метода Данилевского.

Может возникнуть ситуация, когда элемент, относительно которого совершаются очередные преобразования матрицы, равен нулю. Пусть на некотором шаге пришли к виду:

, причем .

а) Если в ()-й строке левее элемента есть ненулевой элемент (,), то меняем местами ()-й и -й столбцы и ()-ю и -ю строки для сохранения подобия. Теперь элемент и можно продолжать преобразования.

б) Если все элементы ()-й строки левее элемента равны нулю, то матрица будет иметь вид:

.

В этом случае характеристический определитель матрицы будет равен произведению определителей матриц и :

При этом матрица уже имеет форму Фробениуса, поэтому метод Данилевского применяют в дальнейшем только к .

Рассмотрим метод Данилевского более подробно для матриц второго и третьего порядка.

1. . Рассматриваем матрицу . Ищем подобную ей матрицу вида .

,

Пусть , тогда: , .

Характеристический многочлен:

.

2. . Рассматриваем матрицу . Ищем подобную ей матрицу вида . Пусть ,

|  |  |  |  |
| --- | --- | --- | --- |
|  | ;  , |  | |
|  |  | | . |

Пусть

|  |  |  |
| --- | --- | --- |
|  | ;  , |  |
|  |  | |

Характеристический многочлен:

Самостоятельно разобрать поэтапно метод Данилевского для матрицы четвертой степени.

ПРИМЕРЫ.

**Пример 1**. Дана матрица , найти ее собственные числа и собственные векторы.

*1. Метод непосредственного развертывания*

Характеристический многочлен:

Корни характеристического многочлена – собственные числа матрицы : и .

Рассмотрим и найдем собственный вектор, соответствующий .

.

, отсюда , значит , например или любой другой коллинеарный вектор.

, найдем собственный вектор, соответствующий .

, отсюда , значит , например или любой другой коллинеарный вектор.

*2. Метод Крылова*

, , .

, .

Получаем систему уравнений:

, , .

Характеристический многочлен: . Его корни: и .

Находим коэффициенты по схеме Горнера (, ), можно записать в таблицу:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | |  |  |  |
|  | |  |  |  |
|  | |  |  |  |
|  |  | 1 | –5 | 0 |
|  |  | 1 | –2 | 0 |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  | 1 | –5 | 0 |
|  | 1 | –2 | 0 |

Считаем собственные векторы: , .

*3. Метод Данилевского*

, тогда: , .

,

Характеристический многочлен: . Его корни: и .

, , .

, , .

**Пример 2**. Дана матрица , найти ее собственные числа и собственные векторы.

*1. Метод непосредственного развертывания*

Корни характеристического многочлена – собственные числа матрицы : , и .

:

|  |  |
| --- | --- |
|  | Первое и второе уравнение линейно-зависимые. второе можем не рассматривать. |
|  | Умножим первое уравнение на 3 и вычтем его из второго уравнения |
|  |  |

, ,

:

|  |  |
| --- | --- |
|  | Третье уравнение является разностью между вторым и первым, его можно исключить.  Умножим первое уравнение на 2 и прибавим ко второму. |
|  |  |

, ,

*2. Метод Крылова*

, ,

, .

. .

Получаем систему уравнений:

, ,

, , .

. Корни: , .

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  |  |  |
|  | |  |  |  |  |
|  |  | 1 | –2 | –3 | 0 |
|  |  | 1 | 2 | 1 | 0 |

.

.

*3. Метод Данилевского*

,

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

. Корни: , .



***6.2. Задание к лабораторной работе***

Используя метод Крылова, найти собственные числа и собственные векторы матрицы. Собственные числа определить с четырьмя верными цифрами, а собственные векторы – с тремя десятичными знаками. Проверить полученные значения по определению.

|  |  |
| --- | --- |
| 1.  3.  5.  7.  9.  11.  13.  15.  17.  19. | 2.  4.  6.  8.  10.  12.  14.  16.  18.  20. |

***Содержание отчета***

1. Тема и цель лабораторной работы;

2. Вариант задания на лабораторную работу;

3. Краткие теоретические сведения и описание алгоритма работы программы в виде блок схемы;

4. Листинг разработанной программы с подробными комментариями;

5. Результаты работы программы;

6. Выводы.

***Контрольные вопросы***

1. Что такое собственное число матрицы?
2. Что такое собственный вектор матрицы?
3. Что называется характеристическим уравнением матрицы?
4. Что называется характеристическим многочленом матрицы?
5. Что такое характеристическая матрица?
6. Какой метод используется для определения корней характеристического многочлена?
7. Какой метод используется для определения коэффициентов характеристического уравнения?
8. Каким образом определяются координаты собственного вектора?
9. В чем сущность метода Крылова?

**Лабораторная работа №7. Приближенное решение обыкновенных дифференциальных уравнений методом Эйлера, модифицированным**

**методом Эйлера и методом Рунге-Кутта**

**Цель работы**: изучить численные методы и алгоритмы численного решения дифференциальных уравнений

**Постановка задачи**: реализовать изученные алгоритмы численного решения дифференциальных уравнений и провести сравнение методов.

**7.1. Теоретическая часть**

Рассмотрим некоторые численные методы решения *задачи Коши* обыкновенных дифференциальных уравнений первого порядка. Постановка задачи Коши:

Необходимо найти *частное решение* дифференциального уравнения,

удовлетворяющего *начальными условиям,*

***Численное решение*** ДУ заключается в вычислении функции *y(x)* в некоторых заданных точках , лежащих на определенном отрезке, т.е. .

Множество значений ,  в которых определяется значение функции,  называют *сеткой*, на которой определена функция *y(x)*. Сами координаты при этом называют *узлами сетки*. Чаще всего используются *равномерные сетки*, в которых расстояние между соседними узлами постоянно и называется *шагом сетки* *h* или *шагом интегрирования* дифференциального уравнения

Преобразуем уравнение умножением на *dx*

и проинтегрируем левую и правую части между *i*-ым и *i+*1-ым узлами сетки.

(5.1)

Мы получили выражение для построения решения в узле интегрирования через значения *x* и *y* в *i*-ом узле сетки. Но в правой части последнего уравнения есть интеграл от неявно заданной функции, нахождение которого в аналитическом виде в общем случае невозможно. Численные методы решения  ОДУ различным способом аппроксимируют (приближают) значение этого интеграла для построения формул численного интегрирования обыкновенных дифференциальных уравнений (ОДУ).

Из множества разработанных для решения ОДУ первого порядка методов рассмотрим методы Эйлера, Рунге-Кутта и Адамса.

**Методы численного решения дифференциальных уравнений**

**Метод Эйлера**

Наиболее простым способом численного решения задачи Коши для ОДУ первого порядка является метод Эйлера. В его основе лежит аппроксимация производной отношением конечных приращений зависимой (y) и независимой (x) переменных между узлами равномерной сетки:

,

где - это искомое значение функции в точке .

Если преобразовать это уравнение и учесть равномерность сетки интегрирования, получим итерационную формулу:

Очевидно, что для приближенного вычисления интеграла в методе Эйлера используется простейшая формула интегрирования - формула левых прямоугольников. Ошибка метода Эйлера прямо пропорциональна шагу интегрирования *h*.

***Пример 7.1:***

Используя метод Эйлера, построить приближенное решение для следующей задачи Коши:

на интервале [0,1] с шагом 0,1

***Решение:***

Для первых трех узлов сетки получим:

Результаты вычислений приведены в таблице. Во второй колонке таблицы для сравнения приведены точные значения решения в узлах сетки.

Таблица 7.1. Решение ОДУ методом Эйлера

|  |  |  |
| --- | --- | --- |
| x | Точное решение | Приближенное решение |
| 0 | 0,000000 | 0,000000 |
| 0,1 | 0,004837 | 0,000000 |
| 0,2 | 0,018731 | 0,010000 |
| 0,3 | 0,040818 | 0,029000 |
| 0,4 | 0,070320 | 0,056100 |
| 0,5 | 0,106531 | 0,090490 |
| 0,6 | 0,148812 | 0,131441 |
| 0,7 | 0,196585 | 0,178297 |
| 0,8 | 0,249329 | 0,230467 |
| 0,9 | 0,306570 | 0,287420 |
| 1 | 0,367879 | 0,348678 |

**Усовершенствованный метод Эйлера (метод Гюна).**

Точность метода Эйлера можно повысить, если воспользоваться для аппроксимации интеграла в формуле (5.1) более точной формулой интегрирования – формулой трапеций.

Данное выражение является уравнением относительно , решить которое можно каким-либо итерационным методом. Также можно поступить иначе и приблизительно вычислить значение функции в узле с  помощью обычной формулы Эйлера:

Таким образом, для каждого узла интегрирования производится следующая цепочка вычислений

Данный метод называется модифицированным методом Эйлера или методом Гюна. Благодаря более точной формуле интегрирования, погрешность метода Гюна пропорциональна уже квадрату шага интегрирования, т.е. *h2*

***Пример 7.2:***

Используя модифицированный метод Эйлера, построить приближенное решение для следующей задачи Коши:

на интервале [0,1] с шагом 0,1

***Решение:***

и т.д*.*

Результаты вычислений приведены в таблице. Во второй колонке таблицы для сравнения приведены точные значения решения в узлах сетки.

Таблица 7.2. Решение ОДУ модифицированным методом Эйлера

|  |  |  |
| --- | --- | --- |
| x | Точное решение | Приближенное решение |
| 0 | 0,000000 | 0,00000 |
| 0,1 | 0,004837 | 0,00500 |
| 0,2 | 0,018731 | 0,01903 |
| 0,3 | 0,040818 | 0,04122 |
| 0,4 | 0,070320 | 0,07080 |
| 0,5 | 0,106531 | 0,10708 |
| 0,6 | 0,148812 | 0,14940 |
| 0,7 | 0,196585 | 0,19721 |
| 0,8 | 0,249329 | 0,24998 |
| 0,9 | 0,306570 | 0,30723 |
| 1 | 0,367879 | 0,36854 |

Отметим существенное увеличение точности вычислений по сравнению с методом Эйлера.

**Методы Рунге-Кутты**

Воспользовавшись [формулой Симпсона](http://www.physchem.chimfak.rsu.ru/Source/NumMethods/Integration.php#Simpson_form) для вычисления интеграла в (5.1), можно получить еще более точную формулу для решения задачи Коши для ОДУ первого порядка -  широко используемого в вычислительной практике метода Рунге-Кутты.

В [формуле Симпсона](http://www.physchem.chimfak.rsu.ru/Source/NumMethods/Integration.php#Simpson_form) для приближенного вычисления определенного интеграла используются значения подынтегрального выражения в  трех точках. В интеграле их всего две, поэтому введем дополнительную точку в середине отрезка [xi+1, xi]

тогда:

Полученное выражение является неявным, так как в правой части содержатся  еще не определенные значения функции *yi+h/2* и *yi+1*. Чтобы воспользоваться этой формулой, надо использовать некоторое приближение для вычисления этих значений.

При использовании различных методов приближенного вычисления этих величин, получаются выражения для методов Рунге-Кутты различного порядка точности.

*Алгоритм Рунге-Кутты третьего порядка -* **РК3** *(погрешность порядка h3)*:

, где

*)*

*Алгоритм Рунге-Кутты четвертого порядка-* **РК4** *(погрешность порядка h4)*:

, где

*)*

Алгоритмы третьего и четвертого порядков требуют на каждом шаге трех и четырех вычислений функции соответственно, но являются весьма точными.

***Пример 7.3.:***

Используя алгоритм Рунге-Кутты третьего и четвертого порядков решить задачу Коши

на интервале [0,1] с шагом 0,1

***Решение:***

Для алгоритма третьего порядка для узла *x*=0.1 вычисления таковы:

*)=*0,1(0-0)=0

*=*0,1[(0+0,1)+(0+2\*0,005-0)]=0,009

=0,0048333

Для алгоритма четвертого порядка для узла *x*=0.1 вычисления таковы:

*)=*0,1 (0-0)=0

,

Приведем таблицу решения с шагом интегрирования h=0,1 методов Рунге-Кутты 3-го (РК3) и четвертого (РК4) порядков на интервале [0,1].

Таблица 5.3. Решение ОДУ методами РК3 и РК4

|  |  |  |  |
| --- | --- | --- | --- |
| x | Точное решение | Метод РК3 | Метод РК4 |
| 0 | 0,000000 | 0 | 0 |
| 0,1 | 0,004837 | 0,00483333 | 0,0048375 |
| 0,2 | 0,018731 | 0,01872336 | 0,0187309 |
| 0,3 | 0,040818 | 0,04080819 | 0,04081842 |
| 0,4 | 0,070320 | 0,07030794 | 0,07032029 |
| 0,5 | 0,106531 | 0,10651697 | 0,10653093 |
| 0,6 | 0,148812 | 0,14879677 | 0,14881193 |
| 0,7 | 0,196585 | 0,19656961 | 0,19658562 |
| 0,8 | 0,249329 | 0,24931274 | 0,24932929 |
| 0,9 | 0,306570 | 0,30655314 | 0,30656999 |
| 1 | 0,367879 | 0,36786283 | 0,36787977 |

Высокая точность, вместе с достаточной простотой реализации делает метод Рунге-Кутты четвертого порядка одним из самых распространенных численных методов решения задачи Коши ОДУ и систем ОДУ первого порядка.

**7.2. Задание к лабораторной работе**

Используя метод Эйлера и метод Эйлера с пересчетом, решить задачу Коши на отрезке *[a,b]* с шагом *h=0.1*. Проверить полученные значения, используя метод Рунге-Кутты 4 порядка

|  |  |
| --- | --- |
| 1. |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

***Содержание отчета***

1. Тема и цель лабораторной работы;

2. Вариант задания на лабораторную работу;

3. Краткие теоретические сведения и описание алгоритма работы программы в виде блок схемы;

4. Листинг разработанной программы с подробными комментариями;

5. Результаты работы программы;

6. Выводы.

***Контрольные вопросы***

1. В каком виде дается решение задачи Коши численными методами?
2. Что является решением дифференциального уравнения?
3. Как численно решить дифференциальное уравнение методом Эйлера?
4. Как численно решить дифференциальное уравнение методом Рунге-Кутты?

**Лабораторная работа №8. Приближенное решение обыкновенных дифференциальных уравнений методом Адамса**

**Цель работы**: изучить численные методы и алгоритмы численного решения дифференциальных уравнений

**Постановка задачи**: реализовать изученные алгоритмы численного решения дифференциальных уравнений и провести сравнение методов.

**8.1. Теоретическая часть**

**Многошаговые методы решения ОДУ.** **Метод Адамса**

Рассмотренные ранее методы (Эйлера, Рунге-Кутты) используют значение функции на одном предшествующем шаге, поэтому они относятся к одношаговым методам. Точность вычислений можно увеличить, если использовать при нахождении  решения в некотором узле *xi* информацию о значениях функции, полученных в нескольких (*k*) предыдущих узлах сетки интегрирования *(xi-1, xi-2 … xi-k).*

Если используются значения в *k* предыдущих узлах, то говорят о *k*-шаговом методе интегрирования ОДУ. Суть его заключается в следующем: по значениям функции, вычисленным в *k* предшествующих узлах, строится интерполяционный полином Лагранжа степени *(k-1)* - , который используется при интегрировании ОДУ. Интеграл при этом выражается через квадратурную формулу:

где квадратурные коэффициенты

Очевидно, что при *k=1* в качестве частного случая получается уже известная  [формула Эйлера](http://www.physchem.chimfak.rsu.ru/Source/NumMethods/ODE.html#6-5). Значения квадратурных коэффициентов для k от 2 до 4 приведены в таблице.

Полученное таким образом семейство формул называется явной k-шаговой схемой Адамса (метод Адамса-Башфорта).

Например, четырех шаговая явная формула Адамса может быть записана так:

Если для построения интерполяционного полинома использовать k узлов, начиная с xi+1, то можно получить формулы интегрирования ОДУ, известные как неявные схемы Адамса (или методы Адамса-Моултона).

Например, четырех шаговая неявная формула Адамса-Моултона имеет вид:

Достоинством многошаговых методов Адамса при решении ОДУ заключается в том, что в каждом узле рассчитывается только одно значение правой части ОДУ - функции *F(x,y).* К недостаткам можно отнести невозможность старта многошагового метода из единственной начальной точки, так как для вычислений по *k*-шаговой формуле необходимо знание значения функции в *k* узлах. Поэтому приходится *(k-1)* решение в первых узлах x1, x2, …, xk-1 получать с помощью какого-либо одношагового метода, например метода [Рунге-Кутты 4–го порядка](http://www.physchem.chimfak.rsu.ru/Source/NumMethods/ODE.html#RK4_form).

**8.2. Задание к лабораторной работе**

* Используя метод Адамса с третьими разностями решить задачу Коши на отрезке [0,1] c шагом h=0,1. Все вычисления вести с четырьмя знаками. Начальный отрезок определить методом Рунге-Кутты. Проверить полученные значения, используя метод Эйлера с пересчетом.

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

***Содержание отчета***

1. Тема и цель лабораторной работы;

2. Вариант задания на лабораторную работу;

3. Краткие теоретические сведения и описание алгоритма работы программы в виде блок схемы;

4. Листинг разработанной программы с подробными комментариями;

5. Результаты работы программы;

6. Выводы.

***Контрольные вопросы***

* 1. В каком виде дается решение задачи Коши численными методами?
  2. Что является решением дифференциального уравнения?
  3. Как численно решить дифференциальное уравнение методом Адамса?

# **Список литературы**

1. Демидович Б.П. Основы вычислительной математики: Учебное пособие. 6-е издание/Б.П. Демидович, И.А. Марон. - СПб.: Издательство «Лань», 2007. -672 с.