**Методические указания к практическим занятиям**

**«Методы математического анализа геосистем»**

|  |  |
| --- | --- |
| Лабораторное занятие №1 | **Точные методы решения СЛАУ.** Некоторые сведения из теории погрешностей в вычислительной математике. Прямые методы решения СЛАУ. Метод Гаусса. |

Цель работы.

Привести некоторые сведения из теории погрешностей в вычислительной математике. Изучить метод Гаусса с выбором главного элемента для численного решения систем линейных алгебраических уравнений (СЛАУ). Научиться программировать алгоритмы решения СЛАУ. Применить метод Гаусса с выбором главного элемента для вычисления определителя матрицы и обратной матрицы.

###### Некоторые сведения погрешности вычисления.

Погрешность приближенного числа *а,* т. е. разность *а* — *а0* между ним и точным значением *а0*, обычно неизвестна.

Величину

∆(𝑎) = |𝑎 − 𝑎0|

называют *абсолютной погрешностью* числа *а.*

*Относительной погрешностью приближенного числа а* называется отношение его абсолютной погрешности к абсолютной величине числа *а,* т. е.

∆(𝑎)

Вычисления с учетом погрешностей:

𝛿𝛿(𝑎) =

|𝑎|

∆(𝑐𝑐𝑎) = 𝑐𝑐∆(𝑎)*,* где 𝑐𝑐 − точное число (константа);

∆(𝑎 ± 𝑏) = ∆(𝑎) ± ∆(𝑏)

∆(𝑎𝑏) = |𝑎| ∆(𝑏) + |𝑏| ∆(𝑎)

𝑎

∆ � � =

𝑏

|𝑎| ∆(𝑏) + |𝑏| ∆(𝑎)

𝑏2

|𝑎| 𝛿𝛿(𝑏) + |𝑏| 𝛿𝛿(𝑎)

𝛿𝛿(𝑎 ± 𝑏) =

𝑎

|𝑎 ± 𝑏|

𝛿𝛿(𝑎𝑏) = 𝛿𝛿 �

𝑏

� = 𝛿𝛿(𝑎) + 𝛿𝛿(𝑏)

###### Теоретические сведения. Метод Гаусса с выбором главного элемента. Вычисление определителя и обратной матрицы.

* 1. Основная идея метода.

Пусть дана система линейных алгебраичес ких уравнений

*Ax = b,* (1)

где *А = (а*ij*)* - неособенная размерности *n**n* матрица, *detA*  0, *x =* (*x*1*, ..., xn*)T и *b =*(*b*1*, ..., bn*)T -

векторы-столбцы.

Вычислительная схема метода состоит из двух этапов. Первый этап заключается в приведении системы к верхнетреугольному виду. Этот этап называется прямым ходом. Второй этап - определение неизвестных x1, x2,..., xn называется обратным ходом.

Прямой ход состоитв последовательном исключении (обнулении) коэффициентов при неизвестных x1, x 2, ..., xn-1, начиная с первого столбца. Исключение коэффициентов при xk-ой переменной называется k -ым циклом метода исключения Гаусса.

Прямой ход реализуется по следующим формулам (индекс k в круглых скобках означает номер цикла (номер столбца) ).

Умножение k - ой строки на число



Вычитание k - ой строки из m - ой строки

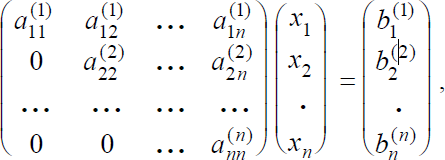




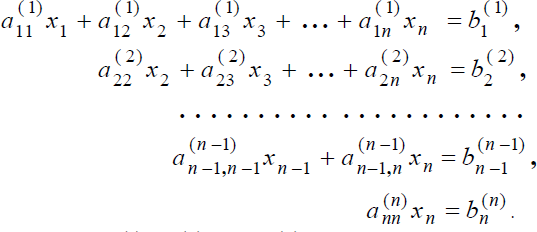
(2)

(3)

(4)



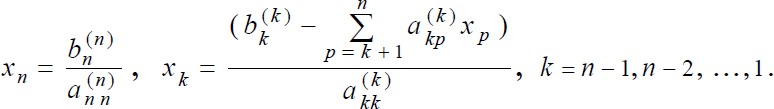
(5)



Элементы, стоящие на главной диагонали полученой системы с верхнетреугольной матрицей (5), называются ведущими элементами.

Обратный ход - вычисление неизвестных xn, xn-1, ..., x1 реализуется по следующим

формулам, начиная с последнего уравнения системы (5)



(6)

Формулами (2)-(6) определяется поэлементная форма записи *основной вычислительной схемы метода исключения Гаусса*. Необходимым и достаточным условием применимости

метода является неравенство нулю всех *ведущих элементов*

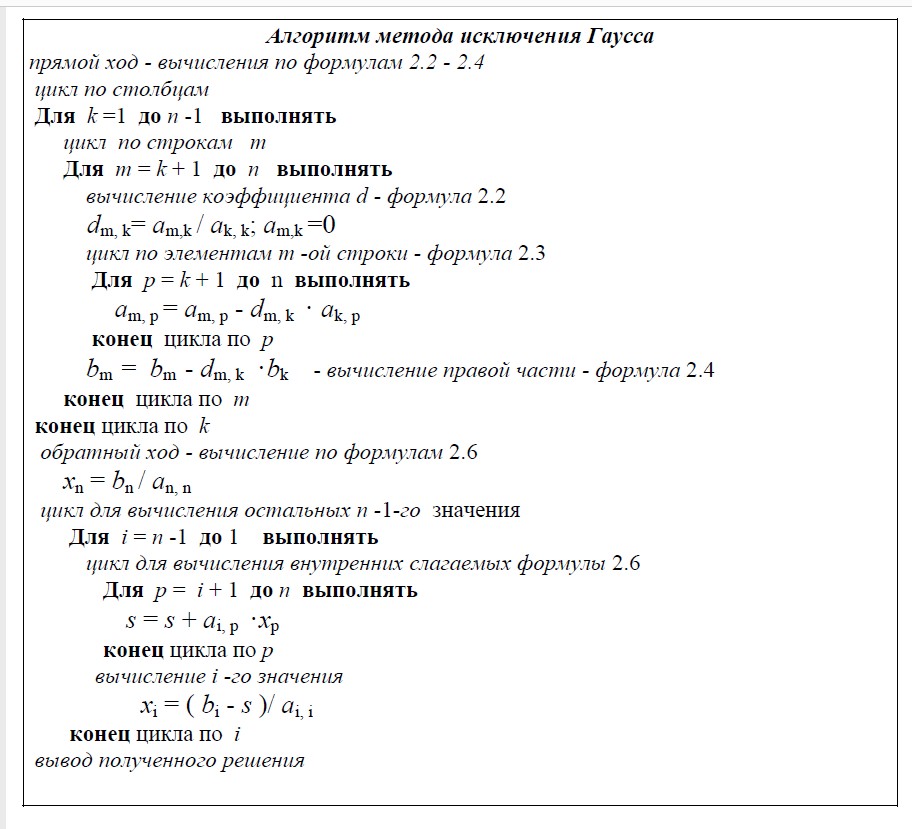


Рисунок 1 - Описание алгоритма основной схемы метода исключения Гаусса

###### Вычисление определителя и обратной матрицы методом Гаусса.



(7)

**Вычисление обратной матрицы.** Применение метода исключения Гаусса для нахождения матрицы, обратной к исходной матрице A, основывается на том, что обратная матрица A-1 является единственным решением матричного уравнения

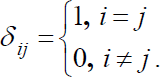
AX = E, (8)

где E - единичная матрица, X – n\*n - матрица.

Если обозначить A-1 = X, то матрица X с элементами xij становится искомой. Перемножение матриц A и X приводит к n системам уравнений относительно n2 неизвестных xij



(9)

где

Выражение (9) определяет систему уравнений для вычисления обратной матрицы в поэлементной форме. Если обозначить столбцы матрицы X, как векторы xi = (x1i, x 2i, ..., x ni)T,

Ax1 = e1 ; Ax2 = e2 ; ... ; Axn = en, (10)

с одной и той же матрицей A и правыми частями, отличающимися только расположением 1 в элементах векторов ei. Для решения каждого из уравнений (10) применяется вычислительная схема (2)-(6), а получаемые значения xi будут составлять столбцы обратной матрицы A-1 .

В реальных вычислениях используются методы с ***выбором главного*** (или **ведущего**)

###### элемента:

* 1. Выбор главного элемента по столбцам.
  2. Выбор главного элемента по строкам.
  3. Выбор главного элемента по всей матрицу

*Выбор главного элемента* **по столбцам** реализуется следующим образом: перед исключением x1 отыскивается

𝑚𝑎𝑥𝑖𝑖=1,…𝑛|𝑎𝑖𝑖1|

Пусть максимум достигается при *i=k*. В этом случае меняются местами первое и k уравнения (или в матрице меняются местами две строки) и реализуется процедура исключения.

Затем отыскивается

𝑚𝑎𝑥𝑖𝑖=1,…𝑛

(1)

𝑖𝑖2

|𝑎 |

и процедура поиска *главного элемента* в столбцах повторяется. Так же реализуется *выбор главного элемента* по строкам: перед исключением *x1* отыскивается

𝑚𝑎𝑥𝑗𝑗=1,…,𝑛|𝑎𝑘𝑗𝑗|

Если максимум достигается при i = k, то у u1 и uk меняются номера, то есть максимальный элемент из коэффициентов первого уравнения окажется на месте a11, и т.д.

Наиболее эффективным является *метод Гаусса с выбором главного элемента* по всей матрице. Во многих методах важным является условие **диагонального преобладания**

|𝑎𝑖𝑖𝑖𝑖| ≥ ∑𝑛

𝑗𝑗=1

𝑗𝑗≠𝑖𝑖

|𝑎

𝑖𝑖𝑗𝑗|

для i = 1, …, n, при выполнении которого проблемы, появляющиеся в *методе Гаусса*, не возникают. Если для всех строк матрицы выполняются строгие неравенства, то говорят о строгом диагональном преобладании.

###### Блок схема метода Гаусса с выбором главного элемента

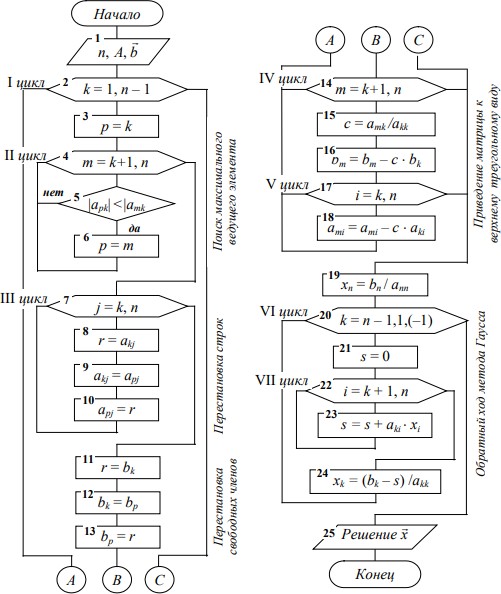


Рисунок 1. — Блок-схема метода Гаусса с выбором главного элемента

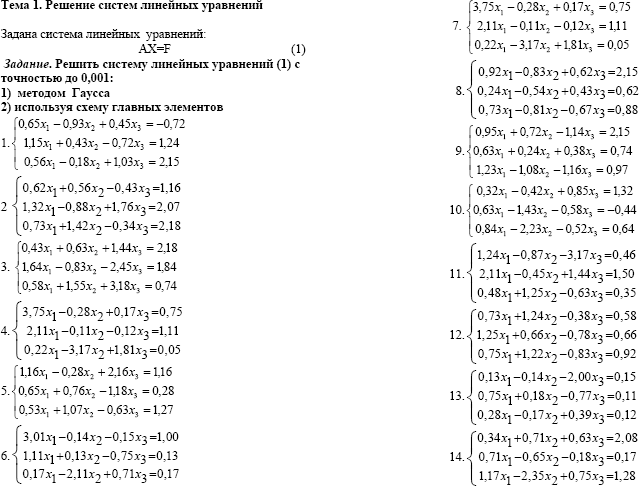
###### Задание.

1. Решить систему линейных алгебраических уравнений методом Гаусса с выбором главного элемента. Варианты задания приведены.
2. Найти определитель и обратную матрицу системы линейных алгебраических уравнений методом Гаусса с выбором главного элемента. Варианты задания приведены.

###### Порядок выполнения работы.

* 1. Разработать алгоритм решения.
  2. Написать программу для ЭВМ.
  3. Произвести расчеты на ЭВМ.
  4. Оформить отчет: а) теоретическая часть; б) вычислительный алгоритм; в) текст программы; г) результаты.

###### Варианты задания.



|  |  |
| --- | --- |
| Лабораторное занятие №2 | Вычислительный алгоритм метода LU разложения и метода Холецкого (метод квадратного корня) для решения СЛАУ |

***Цель работы*:** изучение метода LU и метода квадратного корня ля численного решения СЛАУ, получение практических навыков построения алгоритмов решения систем линейных алгебраических уравнений, программной реализации их на компьютере.

вид

Система линейных алгебраических уравнений (СЛАУ) в самом общем случае имеет

*a*11*x*1  *a*12 *x*2  *a*1*n xn*  *b*1 *a*22 *x*1  *a*22 *x*2  *a*2*n xn*  *b*1

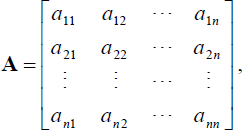
… …

*an*1*x*1  *an* 2 *x*2  *ann xn*  *bn*

(1)

где a ij заданные коэффициенты системы, bj известные свободные члены системы, xj неизвестные, подлежащие определению.

Вводя матрицу коэффициентов системы



вектор-столбец свободных членов



и вектор-столбец неизвестных



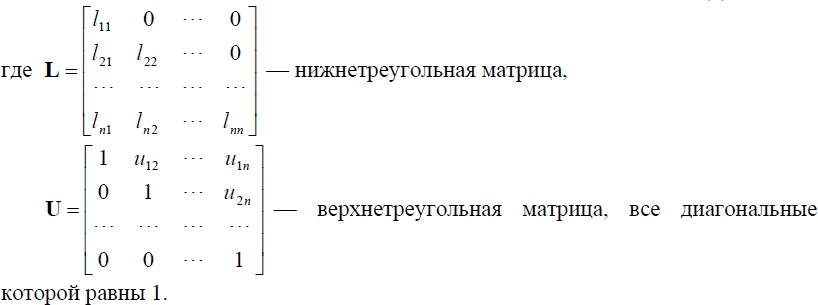
систему (1) можно представить в матричной форме

**AX=B** (2)

Наиболее эффективные прямые методы решения СЛАУ используют разложения матрицы A.

В методе LU-разложения матрица коэффициентов А представляется в виде произведения матриц L и U.

**A**=**LU** (3)



Вектор В в ходе разложения не изменяется.

###### Вычислительный алгоритм метода LU для решения СЛАУ.

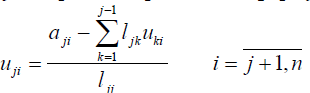
Метод LU-разложения состоит из двух этапов:

1. факторизация (разложение) матрицы А;
2. получение решения Х.

Факторизация матрицы А выполняется за n стадий. На каждой j-той стадии последовательно пересчитываются элементы lij очередного j-го столбца матрицы L по формулам

(4)

и элементы uji очередной j-й строки матрицы U по формулам

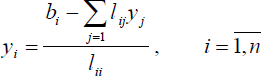


(5)

Получение решения Х системы (2) производится следующим образом: сначала решается система с нижнетреугольной матрицей L вида

**LY**=**B**

с помощью алгоритма прямого хода:

 (6)

Затем решается система UX=Y с верхнетреугольной матрицей U по формуле обратного хода

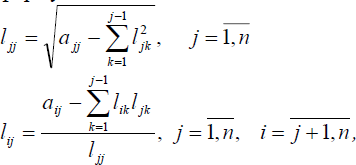
(7)

###### Вычислительный алгоритм метода Холецкого (метод квадратного корня) для решения СЛАУ.

Если матрица A симметрична и положительно определена, то ее факторизуют в виде:

**A***=***LL**T

Для этого на каждой j-й стадии факторизации матрицы A вычисляют элементы lij

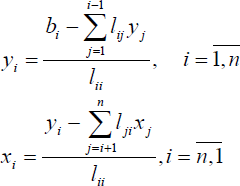


Как видно из приведенных выражений для факторизации матрицы A необходимо извлекать квадратный корень. Поэтому этот метод еще называют методом квадратного корня.

После факторизации матрицы A решение задачи (2) сводится к решению двух СЛАУ с треугольными матрицами

**LY***=***B***,* **L**T**X***=***Y**

Решение этих систем находят по рекуррентным формулам



###### Порядок выполнения работы.

1. Изучить метод LU разложения и метод квадратного корня.
2. Разработать алгоритм решения.
3. Написать программу для ЭВМ.
4. Произвести расчеты на ЭВМ.
5. Оформить отчет: а) теоретическая часть; б) вычислительный алгоритм; в) текст программы; г) результаты.

|  |  |
| --- | --- |
| Лабораторное  занятие №3 | Вычислительный алгоритм и численная реализация метода прогонки для  решения СЛАУ |

1. Описание метода. Вычислительный алгоритм метода прогонки.

Систему уравнений ~Ax=F  c треугольной матрицей

𝐶1

⎛𝐴2

…

𝐴 = ⎜ …0

0

𝐵1 0

𝐶2𝐵2

… …

0 0

… …

. . . .

0 0

0 0

… …

𝐴𝑖𝑖 𝐶𝑖𝑖

… …

0

0

…

𝐵𝑖𝑖

…

0

… 0

… …

… …0

𝐴 𝐶

0

0

… ⎞

…0 ⎟

𝐵

𝗁 0

0

0 0 0 0 0

𝑛−1

𝑛−1

𝐴𝑛

𝑛−1

𝐶𝑛 ⎠

можно записать в координатной форме

~A_{i}x_{i-1}+C_{i}x_{i}+B_{i}x_{i+1} = F_{i}.\qquad\qquad(1) 

Где

#### i=1,…,n-1

𝐴1 = 𝐵𝑛 = 0.

Метод прогонки основывается на предположении, что искомые неизвестные связаны рекуррентным соотношением:

x_i = \alpha_{i+1}x_{i+1} + \beta_{i+1},\,\! ~i=n-1,n-2,\dots,1.\qquad\qquad(2) 

Метод прогонки состоит из двух этапов:

1. прямая прогонка;
2. обратная прогонка.

**Прямая прогонка**

 \begin{cases} \alpha_{i+1} = \frac{-B_i}{A_i\alpha_i + C_i} \\ \beta_{i+1} = \frac{F_i - A_i\beta_i}{A_i\alpha_i + C_i}\end{cases} 

#### i=2,…,n-1

Из первого уравнения получим:

\begin{cases} \alpha_2 = \frac{-B_1}{C_1} \\  \beta_2 = \frac{F_1}{C_1}\end{cases} 

i=n-1..1 \,\! 



###### Достаточное условие применимости метода прогонки.

Если выполняются следующие условия

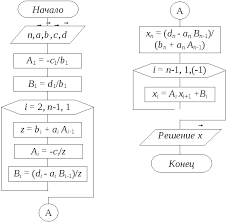
|𝐶𝑖𝑖| ≥ |𝐴𝑖𝑖| + |𝐵𝑖𝑖|, 𝑖𝑖 = 1, … , 𝑛 (\*)

где

𝐴1 = 𝐵𝑛 = 0, 𝐴𝑖𝑖 ≠ 0, 𝑖𝑖 = 2, … , 𝑛; 𝐵𝑖𝑖 ≠ 0, 𝑖𝑖 = 2, … , 𝑛 − 1;

и хотя бы одно неравенство в (\*) строгое. Тогда метод прогонки применим к исходной системе.

###### Блок-схема метода прогонки



**Порядок выполнения работы.**

1. Написать вычислительный алгоритм метода прогонки для решения СЛАУ с трехдиагональной матрицей .
2. Написать программу решения СЛАУ методом прогонки. Решить с ее помощью СЛАУ из своего варианта.
3. Оформить отчет: а) теоретическая часть; б) вычислительный алгоритм метода прогонки; в) текст программы; г) результаты.

|  |  |
| --- | --- |
| Лабораторное занятие №4 | Вычислительный алгоритм и численная реализация МПИ и метода Якоби. |

Цель работы.

Изучить итерационные методы решения для численного решения СЛАУ: метод простых итераций и метод Якоби. Научиться программировать алгоритмы решения

###### Теоретические сведения. Метод простых итераций и метод Якоби.

**Метод простых итераций**

 (1)

Предположим, что (1) можно заменить равносильной системой вида

  (2)

Итерационная схема при заданном произвольно  имеет следующий вид

 *p* = 0, 1, … (3)

при заданном произвольно . Итерационная схема (3) называется методом промтыъ\*х итераций.

Ниже приведены условия, при которых последовательность (3) сходится к решению системы

(1).

Теорема 1. (Достаточное условие сходимости). *Пусть в * *фиксирована некоторая норма,*

*причем соответствующая норма оператора* **B** *равносильной системы* (2) *оказалась меньше единицы*:

Тогда система (1) имеет одно и только одно решение **x**; при

любом  из  последовательность (3) сходится к решению **x**, причем погрешность p-го приближения (или p-й итерации)

удовлетворяет оценке



###### Метод Якоби.

Разложим матрицу **А** следующим образом

###### A=L+D+U,

где **L –** нижняя треугольная матрица, **D –** диагональная матрица**, U –** верхняя треугольная матрица.

Систему (1) перепишем в следующем виде

или

###### +Dx+Ux=f

**= - (L +U)x+f.** (4)

Рассмотрим итерационный метод

**Dx(k+1) = -(L +U)x(k)+f.** (5)

=0,1,…

**x(0) –** задано.

Из (5) сдедует

если

**x (k+1) = - D-1(L +U)x(k)+ D-1f.** (5)

k=0,1,…

𝑎𝒊𝒊𝒊𝒊 ≠ 𝟎𝟎, 𝒊𝒊 = 1, … , 𝑛.

Отметим, что (5) – это метод простой итерации.

Таким образом, метод Якоби свели к МПИ (5).

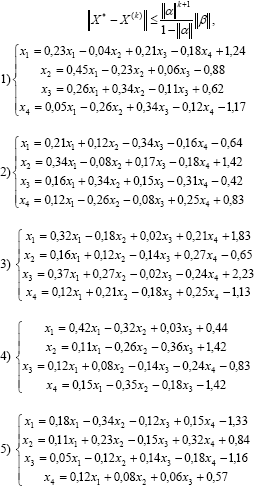
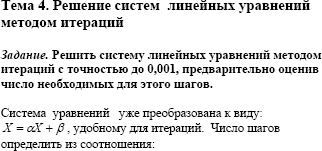
Теорема 2. (Достаточное условие сходимости). Если **А** – матрица с преобладанием главных элементов, то метод Якоби сходится к единственному решению системы (1) при любом  из .

* 1. **Задание.** Решить СЛАУ методом простых итераций и методом Якоби. Варианты задания приведены.

###### Порядок выполнения работы.

1. Изучить метод простых итераций и метод Якоби.
2. Разработать алгоритм решения.
3. Написать программу для ЭВМ.
4. Произвести расчеты на ЭВМ.
5. Оформить отчет: а) теоретическая часть; б) вычислительный алгоритм; в) текст программы; г) результаты.

###### Варианты задания.



|  |  |
| --- | --- |
| Лабораторное занятие №5 | Вычислительный алгоритм и численная реализация метода Зейделя и метода релаксации для решения СЛАУ. |

Цель работы.

Изучить итерационные методы решения для численного решения СЛАУ: метод Зейделя и метод релаксации. Научиться программировать алгоритмы решения.

###### Теоретические сведения. Метод простых итераций и метод Якоби.

**Метод простых итераций**

Рассмотрим систему линейных уравнений



Разложим матрицу **А** следующим образом

(1)

###### A=L+D+U,

где **L –** нижняя треугольная матрица, **D –** диагональная матрица**, U –** верхняя треугольная матрица.

Предположим, что 𝑎𝒊𝒊𝒊𝒊 ≠ 𝟎𝟎, 𝒊𝒊 = 1, … , 𝑛., тогда существует обратная матрица **D-1.**

Систему (1) перепишем в следующем виде

###### Lx+Dx+Ux=f.

Рассмотрим итерационный метод

**Lx(k+1) + Dx(k+1) + Ux(k) = f** (2)

k=0,1,…

**x(0) –** задано.

1. называется итерационным методом Зейделя.

Перепишем (2) в виде

**x (k+1) = - D-1Lx(k+1) + D-1Ux(k + D-1f.** (3)

Обозначим

###### B1= - D-1L, B2 =- D-1U, d = D-1f.

Тогда (3) примет вид

**x (k+1) = B1x(k+1) + B2x(k + d.**

* 1. (Достаточное условие сходимости). Пусть выполняется условие

||𝐵𝟐𝟐||

𝟏𝟏 − ||𝐵𝟏𝟏||

≤ 𝒒𝒒 < 𝟏𝟏, 𝑎𝒊𝒊𝒊𝒊 ≠ 𝟎𝟎, 𝒊𝒊 = 1, … , 𝑛

Тогда метод Зейделя сходится к единственному решению системы (1) при любом  из .

###### Метод релаксации.

**(В +** 𝑚**L)( x (k+1) - x (k)) +** 𝑚**A x (k) =** 𝑚**f**

В случае метод называется *методом верхней релаксации*. Обычно полагают Очевидно, что в случае  метод релаксации совпадает с методом Зейделя.



**3. Задание.** Решить СЛАУ методом Зейделя. Варианты задания приведены в семинарской работе №4.

###### Порядок выполнения работы.

1. Изучить метод Зейделя.
2. Разработать алгоритм решения.
3. Написать программу для ЭВМ.
4. Произвести расчеты на ЭВМ.
5. Оформить отчет: а) теоретическая часть; б) вычислительный алгоритм; в) текст программы; г) результаты.

|  |  |
| --- | --- |
| Лабораторное занятие №6 | Метод простой итерации (МПИ) для решения СЛАУ. Численная реализация МПИ для решения нелинейных уравнений и систем нелинейных уравнений. |

Цель работы.

Изучить метод простой итерации для численного решения СЛАУ Научиться программировать алгоритмы для решения линейных систем.

###### Теоретические сведения.

**Метод простой итерации.**

Пусть уравнение (1) приведено к эквивалентному уравнению x = ϕ(x). (2)

𝑥𝑛+1 = 𝜑𝜑(𝑥𝑛), n = 0, 1, 2,... (3)

Если итерационная последовательность {𝑥𝑛} сходится к некоторому пределу, то этот предел есть корень исходного уравнения. Условия сходимости итерационного процесса дает следующая теорема.

**Теорема 1.** Пусть уравнение (2) имеет единственный корень на отрезке [a, b] и выполнены условия:

1. функция ϕ(x) определена и дифференцируема на отрезке [a, b];
2. все значения функции ϕ(x) при x ∈ [a, b] принадлежат этому отрезку;
3. существует число α∈(0,1) такое, что для всех x∈[a, b] имеет место неравенство ⎪ϕ′(x)⎪<

α.

Тогда итерационная последовательность (4) сходится к корню уравнения (3) для любого

x0∈[a, b].

За приближенное значение корня можно взять любой член итерационной последовательности {𝑥𝑛} , n = 0, 1, 2,... При этом погрешность приближения определяется по формуле

|𝑥𝑛+1 − 𝑥∗| ≤

𝛼𝑛 1 − 𝛼

|𝑥1 − 𝑥0|

где x \* — точное решение уравнения (3), а число α определено в теореме 1.

Критерием остановки итерационного процесса является выполнение условия

|𝑥𝑛+1

− 𝑥𝑛

| ≤ 1−𝛼 ≤ ε (5)

𝛼

где ε — заданная точность решения уравнения.

* 1. **Задание.** Решить нелинейные уравнения методом простой итерации. Варианты задания приведены в MOODLE.

###### Порядок выполнения работы.

1. Изучить метод простых итераций для нелинейных уравнений и систем нелинейных уравнений.
2. Разработать алгоритм решения.
3. Написать программу для ЭВМ.
4. Произвести расчеты на ЭВМ.
5. Оформить отчет: а) теоретическая часть; б) вычислительный алгоритм; в) текст программы; г) результаты.

|  |  |
| --- | --- |
| Лабораторное занятие №7 | Вычислительный алгоритм и численная реализация итерационного метода Ньютона для решения нелинейных уравнений и систем нелинейных уравнений. Метод секущих |

Цель работы.

Изучить итерационные методы решения нелинейных уравнений: метод секущих, метод Ньютона. Научиться программировать алгоритмы решения.

###### Теоретические сведения. метод Ньютона.

Пусть дано уравнение

f(x) = 0. (1)

Найти корни этого уравнения с точностью ε > 0. Ограничимся обсуждением методов поиска лишь действительных корней, не затрагивая проблему корней комплексных.

Приближенное решение уравнения (1) обычно разбивается на два этапа:

1. Локализация корней. Отделение корней, т.е. установление промежутков, содержащих по одному корню.
2. Уточнение корней, т. е. сужение отрезка, содержащего корень, до такой степени, что длина отрезка становится меньше требуемой точности.

###### Отделение корней.

В практике распространен графический способ отделения приближенных корней. Принимая во внимание, что действительные корни уравнения (1) — это точки пересечения графика функции f(x) с осью абсцисс, достаточно построить график функции f(x) и отметить точки пересечения f(x) с осью Ох, или отметить на оси Ох отрезки, содержащие по одному корню.

Пример 1. Графическим методом решить уравнение x3 – x2 – 3 = 0.

Р е ш е н и е. Уравнение запишем в виде x3 = x2+ 3. Обозначим f1(x) = x3 , f2(x) = = x2+ 3. Строим графики функций y = x3 и y = x2+ 3 (рис. 1). Находим абсциссу x0 точки пересечения М этих графиков. Получаем x0 ≈1,9, т. е. x0∈[а, b] = [1; 2].

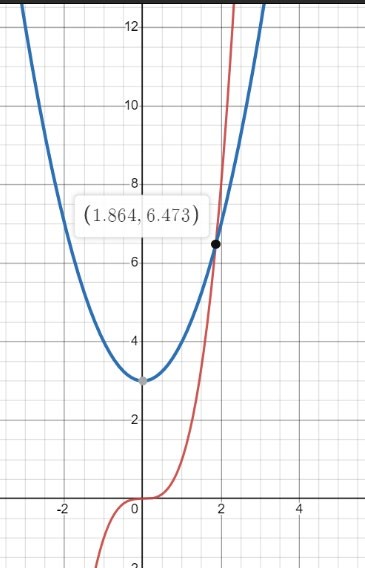
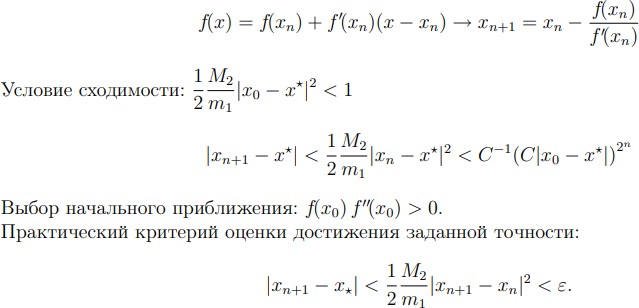


Рис. 1. графики функций y = x3 и y = x2+ 3

###### Метод Ньютона (касательных)



Замечание. Нулевое приближение можно найти графическим путем.

###### Метод секущих.

Данный метод является модификацией метода Ньютона, позволяющей избавиться от явного вычисления производной путем ее замены приближенной формулой. Тогда вместо процесса (2) получаем:

*x*  *x* 

*f* (*xk* 1)*h*

  (*x*

) .*k=1,2,…*

*k k* 1

*f* (*xk* 1)  *f* (*xk* 1

* *h*)

*k* 1

Здесь *h* - некоторый малый параметр метода, который подбирается из условия наиболее точного вычисления производной.

Метод одношаговый *(m=1),* и его условие сходимости при правильном выборе *h* такое же, как у метода Ньютона.

Если f"(x)\*f(a)<0; то x0=a, если же f"(x)\*f(b)<0; то x0=b. X принадлежит интервалу с корнем. Рекуррентное соотношение (3) можно преобразовать к более простой форме.

Каждое последующее приближение вычисляется по рекуррентной формуле

*x*  *x* 

*f* (*xk* 1)

 (*b*  *x*

) *k*  1,2,...

*k k* 1

*f* (*b*) 

*f* (*x*

*k* 1)

*k* 1

, (4)

которая справедлива, если данная функция f(x) на интервале (a;b) выпуклая: ( *f* (*b*) *f*  (*b*)  0 ). -

и *x*0  *a* , Если же функция вогнута, используют следующую рекуррентную формулу:

*x*  *x* 

*f* (*xk* 1)

 (*a*  *x*

) *k*  1,2,..

*k k* 1

*f* (*a*) 

*f* (*x*

*k* 1)

*k* 1

(5)

*f* (*a*) *f*  (*a*)  0

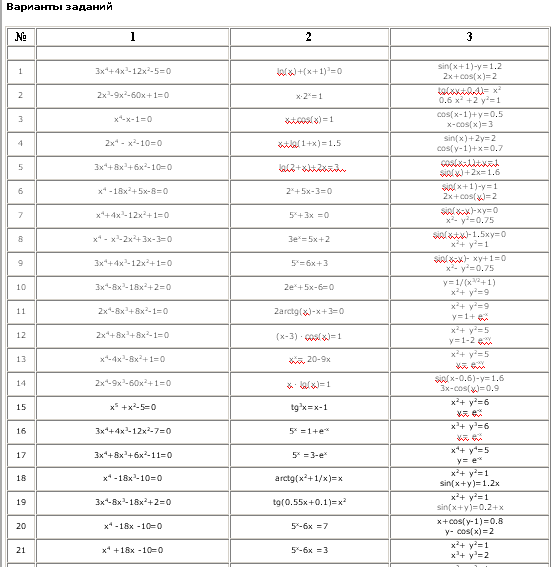
и *x*0  *b*

1. **Задание.** Решить нелинейные уравнения методом Ньютона. Варианты задания приведены.

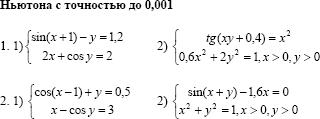
###### Порядок выполнения работы.

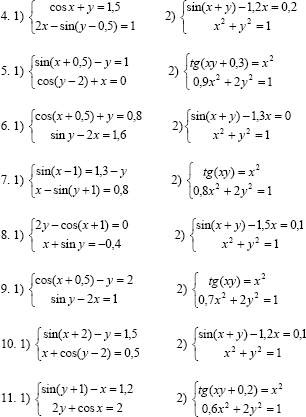
* 1. Изучить итерационный метод Ньютона..
  2. Разработать алгоритм решения.
  3. Написать программу для ЭВМ.
  4. Произвести расчеты на ЭВМ.
  5. Оформить отчет: а) теоретическая часть; б) вычислительный алгоритм; в) текст программы; г) результаты.

###### Варианты задания.









|  |  |
| --- | --- |
| Лабораторное занятие №8 | Интерполирование функций с помощью интерполяционной формулы Лагранжа. Численная реализация |

Цель работы.

Изучить приближенное вычисление первой и второй производных с помощью интерполяционной формулы Лагранжа. Научиться программировать алгоритмы решения.

###### Теоретические сведения.

**Интерполяционная формула Лагранжа.**

**Определение.** Пусть функция f(х) задана таблично на [a,b]:

x0 = a, xn = b , x0 < x1 < x2 < …. < xn , yi = f(xi) i = 0,…, n

Тогда построение непрерывной на [a,b] функции  (x) , такой что  (xi) = yi называется интерполяцией функции f(x) на [a,b].

**Определение.** Пусть полином степени n Ln(x) = a0 xn + a1 xn-1 + … + an интерполирует y=f(x) на [a,b], т.е. Ln(xi) = yi= f(xi). Тогда Ln(x) называется интерполяционным полиномом.

**Утверждение.** Интерполяционный многочлен степени n для функции y=f(x), заданной таблично в n+1 точках, существует и единственен.

Данное утверждение следует из того, что определитель Вандермонда отличен от нуля.

Существуют некоторые стандартные формы записи интерполяционных полиномов. Интерполяционный многочлен в форме Лагранжа имеет вид

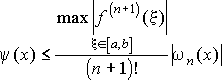
Image70  .

**Теорема.**

Пусть функция y=f(x) имеет n+1 непрерывную производную на [a,b], и Ln(x) –

интерполяционный многочлен, Ln(xi) = f(xi) , i=0,1,…,n. Тогда для погрешности интерполяции

𝝍𝝍(x) = L(x) – f(x) справедлива оценка



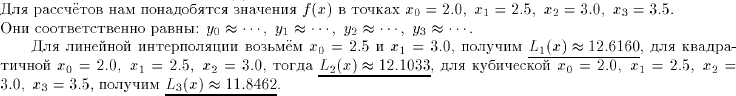
где

Image77 



Пример. Рассмотрим функцию





Таким образом, чем больше узлов интерполирования возьмете, тем точнее получитье значение функции в заданной точке.

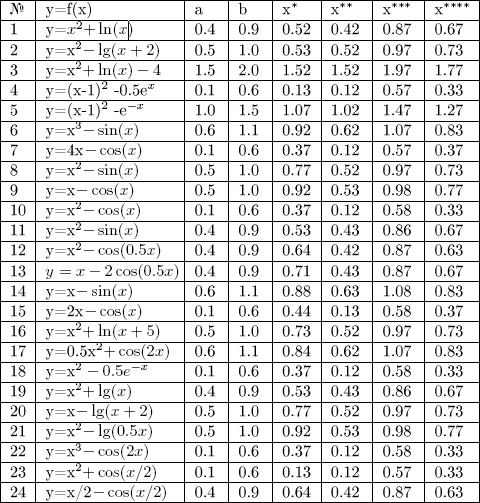
###### Задание. Найти значение функции в заданной точке (n=10).

**Порядок выполнения работы.**

* + - 1. Изучить интерполяционный многочлен Лагранжа.
      2. Разработать алгоритм решения.
      3. Написать программу для ЭВМ.
      4. Произвести расчеты на ЭВМ.
      5. Оформить отчет: а) теоретическая часть; б) вычислительный алгоритм; в) текст программы; г) результаты.

###### Варианты задания.

Таблица 1.



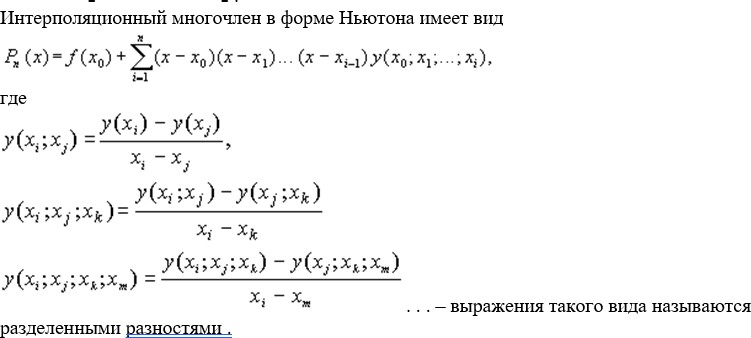
|  |  |
| --- | --- |
| Лабораторное занятие №9 | Интерполяционные формулы Ньютона. Первая и вторая формулы Ньютона |

Цель работы.

Изучить интерполяционные формулы Ньютона: первая и вторая формулы Ньютона. Научиться программировать алгоритмы решения.

1. **Теоретические сведения.** Интерполяционные формулы Ньютона.

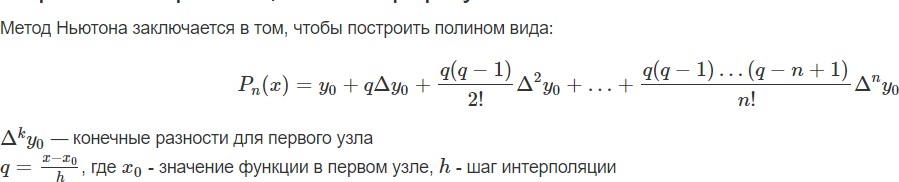
###### Построение интерполяционного полинома Ньютона по заданным значениям функции.



В случае равноотстоящих узлов интерполирования многочлен Ньютона можно записать в следующим образом.

.

###### Первая интерполяционная формула Ньютона.



**Вторая интерполяционная формула Ньютона**. Выполнив подстановку *q* = (*х – хn*)/*h*, получим иную запись второго интерполяционного многочлена Ньютона:

*P* (*x*)  *y*

   *q*   2  *q*  (*q* 1)  ...   *n*  *q*  (*q* 1) … (*q*  *n* 1) .

*n n* 1 2! *n*!

(6)

Первая интерполяционная формула Ньютона используется для интерполирования в начале отрезка [*хi*, *хi*+1] и эктраполирования до первой точки *х*0, т.е. для интерполирования вперед и экс- траполирования назад. При таком интерполировании *q* = (*х – хi*)/*h* > 0. При экстраполировании назад по первой интерполяционной формуле *q* = (*х – хi*)/*h* < 0. При интерполировании в конце таблицы, т.е. при интерполировании назад, когда шаг интерполяции постоянен, применяют вто-

рую интерполяционную формулу Ньютона, где *q* = (*х – хi*+1) / *h* < 0. Эту же формулу используют при экстраполировании вперед (в конце таблицы), но тогда *q* = (*х - -хi*+1)/ *h*>0.

1. **Задание.** Пользуясь первой и второй формулами Ньютона вычислить значение функции в заданной точке.

###### Порядок выполнения работы.

* 1. Изучить интерполяционные формулы Ньютона.
  2. Разработать алгоритм решения.
  3. Написать программу для ЭВМ.
  4. Произвести расчеты на ЭВМ.
  5. Оформить отчет: а) теоретическая часть; б) вычислительный алгоритм; в) текст программы; г) результаты.

Варианты задания приведены в таблице 1 (см. задание №8).

|  |  |
| --- | --- |
| Лабораторное занятие №10 | Квадратурная формула интерполяционного типа. Квадратурная формула Ньютона-Котеса. Оценки погрешности. Квадратурные формулы прямоугольника, трапеции, Симпсона. |

Цель работы.

Изучить квадратурную формулу Ньютона–Котеса, формулу трапеций и ее остаточный член.. Научиться программировать алгоритмы решения.

###### Теоретические сведения. Квадратурная формула Ньютона–Котеса, Формула трапеции.

Пусть некоторая функция *f*(*х*), как и раньше, задана в виде таблицы значений *yi = f*(*хi*) в узлах интерполяции *хi = =х*0 *+ ih* на отрезке [*а, b*]. Требуется найти значения интеграла на указанном отрезке.

По заданным значениям подынтегральной функции *yi = =f*(*хi*) построим интерполяционный полином Лагранжа

,

который для равноотстоящих узлов примет вид

,

где *q* = (*х – х*0) / *h*.

Теперь заменим подынтегральную функцию *f*(*х*) построенным полиномом, считая, что узлы интерполяции расположены равномерно:

*b b b*

*n*

###   

 1*n**i*

*q**q*  1… *q*  *n*

   *f x dx*   *Ln dx* 

*yi*  *i*!*n*  *i*! 

*q*  *i*

*dx*.

*a a a i* 0

Проведя необходимые элементарные преобразования, выполнив замену переменных *dq = dx/n* и сменив в соответствии с подстановкой пределы интегрирования, получим

*n*

*b*

#####   

 1*n**i*

*n*

#####        

0

   *f*

*a*

*x dx* 

*i* 0

*yi*  *i*!*n*  *i*!  *h*   *q q*  1 … 

*q*  *i*  1 

*q*  *i*  1 … 

*q*  *n dq* .

Здесь *h* – шаг, который для равноотстоящих узлов интерполяции определяется как *h =* (*b*

*– а*)*/n*. Подставив значения *h* в последнюю формулу, окончательно получим

,

где

;

*Нi* – *коэффициенты Ньютона – Котеса*. Они не зависят от значений функции *f*(*х*) и являются функциями только количества узлов, на которые разбит отрезок [*а, b*]*.* Поэтому *Нi* обычно вычисляют заранее:

*N =* 1: *Н*0 = *Н*1 = ½;

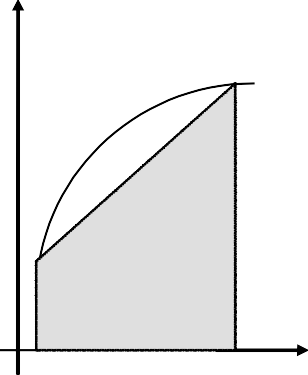
*N =* 2: *Н*0 = *Н*2 = 1/6; *Н*1 = 2/3;

*N =* 3: *H*0 = *H*3 = 1/8; *H*1 = *H*2 = 3/8;

*N =* 4: *H*0 = *H*4 = 7/90; *H*1 = *H*3 = 16/45; *H*2 = 2/13;

###### Квадратурная формула трапеции.

В формуле трапеций используются значения функции в концевых точках эле- ментарных отрезков. В этом случае аппроксимируется площадью трапеции с основаниями *f*(*хi*) и *f*(*хi*+1) и высотой *x* (рис. 1).



f(xi+1)

f(xi)

xi

xi+1

Тогда площадь фигуры может быть определена из формулы площади прямоу- гольной трапеции

*Si =* (*fi + fi+*1) *hi* /2*.*

Если теперь просуммируем последнюю формулу по всем элементарным отрез- кам, то получим с учетом выполненных элементарных преобразований следу- ющее выражение:

*n n*1

Рис. 1 Геомет рическая инт ер-

  *i*  *b*  *a*  *f a*  *fb* / 2   *f* *xi*   *f* *xi* 1 / *n* .

прет ация “ мет ода т рапеций”

*i* 1

*i* 2

Формула для оценки погрешности численного интегрирования методом трапеции имеет

вид:

где

*M*  max

*x*[*a*,*b*]

2

*f*  (*x*).

###### Квадратурная формула Симпсона.

Метод Симпсона часто называют в литературе *методом парабол*. Очевидно, что точность вычислений приближенного интеграла возрастет по сравнению с точностью вычислений, выполненных по формулами трапеций и прямоугольников, если подынтегральную функцию *f*(*х*) заменить на отрезке [*хi,хi*+1] квадратичной параболой, которая в узлах разбиения *хi* принимает зна- чения *f*(*хi*) и при этом *х*0 *= =а; f*(*х*0) *= f*(*а*) *= y*0*; хn = b; f*(*хn*) *= f*(*b*) *= yn* .

Разобьем равномерно отрезок [*а, b*] на *N* элементарных отрезков [*хi,хi*+1] и на каждом из них заменим подынтегральную функцию *f*(*х*) интерполяционным многочленом Ньютона (или Лагранжа, в принципе, без разницы!) второй степени. Тогда для каждого элементарного отрезка [*хi,хi*+1] имеем следующее:

*y*

 

 4*h* 2



 *y*  8*h*3  *y*  4*h*3

 2 *i*   2 *i*  



  

   

2*hyi h* 2

2*h* 2 3

2*h* 2

2 2*hyi*

2*h* ( *yi* )

*h* 2 *yi* / 3

 *h*  *yi*  4 *yi* 1 / 2  *yi* 1 / 6 .

Просуммируем полученное выражение по всем элементарным отрезкам, и если подставим *h* =

=(*b – а*) / *n*, то окончательно получим

*N*

  

*N*

*h*  *y*

* 4 *y*

 *y* / 6  *b*  *a*  *y*

 *y*   4 *y*

*N* 1 

 2



*i i*

*i* 0

*i* 1 / 2

*i* 1

 0 *n*

6*N* 

*i* 0

*i* 1 / 2

*i* 1

*yi* .



Данное выражение называется *формулой Симпсона*. Онo относится к формулам повы- шенной точности и является точной для многочленов второй и третьей степени.

Погрешность формулы Симсона оценивается по формуле:

где

*M*  max

*x*[*a*,*b*]

4

*f* (4) (*x*).



1. **Задание.** Вычислить с помощью формулы трапеции определенный интеграл от заданной функции. Варианты задания приведены.

###### Порядок выполнения работы.

* 1. Изучить квадратурную формулу Ньютона-Котеса, формулу трапеции и ее остаточный член.
  2. Разработать алгоритм решения.
  3. Написать программу для ЭВМ.
  4. Произвести расчеты на ЭВМ.
  5. Оформить отчет: а) теоретическая часть; б) вычислительный алгоритм; в) текст программы; г) результаты.

###### Варианты задания.

Таблица 1

|  |  |  |  |
| --- | --- | --- | --- |
| *N* | *Функция f(x)* | *Интервал* | |
| *a* | *b* |
| 1 | 4*x*  7sin(*x*) | -2 | 3 |
| 2 | *x*2  10sin2 (*x*) | 0 | 3 |
| 3 | ln(*x*)  5cos(*x*) | 1 | 8 |
| 4 | *ex* / *x*3  sin3(*x*) | 4 | 7 |
| 5 | *x*  cos2 (*x*) | 5 | 8 |
| 6 | ln(*x*)  5sin2 (*x*) | 3 | 6 |
| 7 | *x*  5sin2 (*x*) | 1 | 4 |
| 8 | sin2 (*x*)  *x* / 5 | 0 | 4 |
| 9 | *x*3  10*x*2 | -8 | 2 |
| 10 | *x*3  5*x*2 | -2 | 5 |
| 11 | *x*3  6*x*2  0.02*ex* | -5 | 3 |
| 12 | *x*2  5cos(*x*) | -1 | 4 |
| 13 | sin2 (*x*)  3cos(*x*) | 1 | 7 |
| 14 | *x*3  50cos(*x*) | -2 | 5 |
| 15 | 0.1*x*3  *x*2 10sin(*x*) | -4 | 2 |

|  |  |
| --- | --- |
| Лабораторное  занятие №11 | Вычисление производных с помощью интерполяционных формул  Лагранжа, Ньютона и Стирлинга. Погрешность аппроксимации на сетке. |

Цель работы.

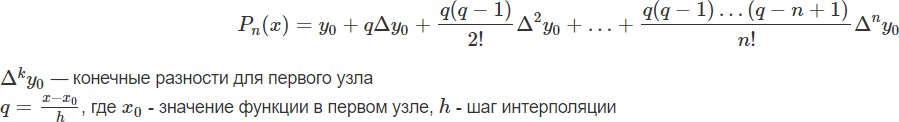
Изучить задачу численного дифференцирования с помощью интерполяционных формул Ньютона и Лагранжа. Научиться программировать алгоритмы решения.

###### Теоретические сведения. Численное дифференцирование с помощью интерполяционных формул Ньютона и Лагранжа.

Пусть некоторая функция *F*(*х*) задана таблично в *n*+1 точке, расположенной на интервале [*а, b*]. Требуется вычислить первую и вторую производные данной функции в заранее определенных точках.

В качестве аппроксимирующей функции выберем интерполяционный многочлен. Если узлы интеpполяции расположены не равномерно, то таким многочленом могут быть полиномы Лагранжа или Лежандра, а если равномерно, то лучше использовать полином Ньютона, так как он дает мень- шую вычислительную погрешность по сравнению с другими.

Пусть узлы *хi*, в которых известны значения функции *F*(*хi*), расположены равномерно на интервале [*а, b*], т.е. *хi+*1*-хi = h*, где *i =* 0, 1*, ..., n.* Тогда в качестве интерполирующей функции *Рn*(*х*) выберем **полином Ньютона,**



Здесь использована запись первой интерполяционной формулы Ньютона. Раскроем скобки, приведем подобные, получим

*f* (*x*)  *y*0  1  *q* 

Заметим, что .

  *q*2  *q* 

2!

2

  *q*3  3*q*2  2*q*

3!

3

+

  *q*4  6*q*3  11*q*2  5*q*

4!

4

+ ...

(1)

Продифференцируем *f*(*х*) из выражения (1) и с учетом последнего замечания получим выражение для первой производной

1    2*q* 1   3*q* 2  6*q*  2   4*q* 3  9*q* 2 11*q*  3 

*f* (*x*)    2  3  4  ... .

*h*

2

6





 1 12

 (2)

Продифференцируем последнее выражение еще раз:

. (3)

Здесь учитывалось, что

.

Аналогично можно определить и следующие производные функции *f*(*х*). Но при вычислении производных более высоких порядков следует учитывать большее количество чле- нов ряда Ньютона для обеспечения точности результата.

Если производные функции *f*(*х*) надо определить в узлах интерполяции, т.е. в *х = хi* , то тогда формулы (2), (3) для первой и второй производных заметно упрощаются, так как в этом случае *q*  0:

*f ’*(*x*) *=* (1 *-* 2/2 *+* 3/3 *-* 4 /4 *+ ...* ) / *h* ; (4)

*f ’’*(*x*) *=* (2 *-* 3 *+*11 4 /12 *-*5 5 / 6 *+ ...* ) / *h*2 *.* (5)

Погрешность в определении производных можно приближенно оценить:

.

Для того чтобы получить значения производных функции *f*(*х*) в точках, расположенных в конце таблицы, следует, воспользоваться второй интерполяционной формулой Ньютона. Повто- рив аккуратно и последовательно все рассуждения, получим для первой производной следующее выражение:

1   *y*  2*q*  1  *y*  3*q* 2  6*q*  2  *y*  2*q* 3  9*q* 2  11*q*  3 

(6)

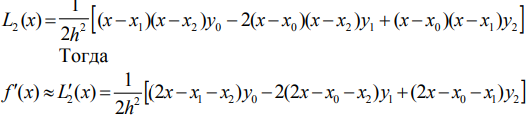
*f* (*x*) 

1 *n* 1

 *y*   2 *n*2  3 *n*3  4 *n*4  ... .

*h*  2 6 12 

На практике часто выгоднее выражать значения производных не через конечные разности, а непосредственно через значения функции в узлах. Для получения таких безразностных формул удобно воспользоваться многочленом Лагранжа с равномерным расположением узлов (*xi-xi-1=h, i=1,2,...,n*).

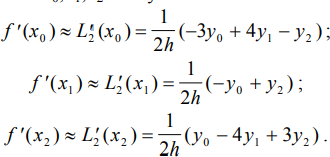
Запишем многочлен Лагранжа второй степени (три узла интерполирования).

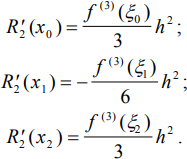
.

Имеет место оценка погрешности интерполирования



Теперь можно вычислить значения производных в трех узлах и их погрешности интерполирования, **используя многочлен Лагранжа:**





В заключение отметим, что рассматриваемые здесь формулы численного дифференцирования являются менее точными по сравнению с интерполяционными. Однако их применяют в практи- ческих расчетах достаточно часто, так как они удобны, просты, не требуют специальных про- грамм. И, что особенно важно, позволяют быстро сделать предварительные (прикидочные) расчеты.

1. **Задание**. Дана таблица значений функции

*y*  *f* (*x*)

. Составить таблицу значений а)

производной точках.

*y*' в заданных точках; б) производной второго порядка

*y*" в заданных

###### Порядок выполнения работы.

* 1. Изучить задачу численного дифференцирования с помощью интерполяционных формул Ньютона и Лагранжа. Разработать алгоритм решения.
  2. Разработать алгоритм решения.
  3. Написать программу для ЭВМ.
  4. Произвести расчеты на ЭВМ.
  5. Оформить отчет: а) теоретическая часть; б) вычислительный алгоритм; в) текст программы; г) результаты.

**Варианты задания.** Задачи №2 а)-в), стр. 131. Н.В. Копченова, И.А. Марон Вычислительная математика в примерах и задачах. – «Наука», М. 1972, 369 с.

|  |  |
| --- | --- |
| Лабораторное занятие №12 | Применение метода конечных разностей для решения задачи Коши для ОДУ первого порядка и систем ОДУ, составление разностной схемы Эйлера. Численная реализация разностной схемы Эйлера. Исследование сходимости схемы Эйлера. |

Цель работы.

Изучить метод конечных разностей для численного решения обыкновенных дифференциальных уравнений (ОДУ) первого порядка. Научиться программировать алгоритмы решения.

###### 1. Теоретические сведения. Построение разностной схемы для ДОУ первого порядка.

Пусть имеется функция f (x), которую необходимо продифференцировать несколько раз и найти эту производную в некоторой точке.

Если задан явный вид функции, то выражение для производной часто оказывается

достаточно сложным и желательно его заменить более простым. Если же функция задана только в некоторых точках (таблично), то получить явный вид ее производных ввобще невозможно. В этих ситуациях возникает необходимость приближенного (численного) дифференцирования.

Простейшая идея численного дифференцирования состоит в том, что функция заменяется интерполяционным многочленом (Лагранжа, Ньютона) и производная функции приближенного заменяется соответствующей производной интерполяционного многочлена

*f* (*m*) (*x*)  *L*(*m*)(*x*),

*n*

*f* (*m*) (*x*)  *l* (*m*)(*x*),0  *m*  *n*.

*n*

Рассмотрим простейшие формулы численного дифференцирования, которые получаются указанным способом.

Будем предполагать, что функция задана в равностоящих узлах

xi 

x0  ih,

h  0,

# i  0,  1,  2,… .

Ее значения и значения производных в узлах будем обозначать

i

i

f ( xi ) 

fi ,

f '( xi ) 

f ',

f ''( xi ) 

f ''.

Пусть функция задана в двух точках и ее значения Рассмотрим интерполяционный многочлен первой степени

l1(x)  f0  (x  x0 ) f (x0;x1).

Производная l' (x) равна

1

l' ( x) 

1

f ( x0; x1 ) 

f1 

# h

f0 .

Производную функцию в точке приближенно заменяем производной интерполяционного многочлена

f ' (x) 

0

f1 

h

f0 .

(1)

Величина называется первой разностной производной. Пусть задана в трех точках

Интерполяционный многочлен Ньютона второй степени имеет вид

l2 (x)  f (x0 )  (x  x0 ) f (x0;x1)  (x  x0 )(x  x1) f (x0;x1;x1).

Берем производную

l' ( x)  f ( x ; x )  (2x  x  x ) f ( x ; x ; x ).

##### 2 0 1 0 1 0 1 1

В точке она равна

l' ( x )  f1  f0  ( x  x ) 

2 0 x1  x0 0 1





( x

f0

 x )( x  x



) ( x

f1

 x )( x  x



) ( x

f1

 x )( x



 x )



 0 1 0 1

1 0 1 1

1 0

1 1 

 f1  f1 .

2h

Получаем приближенную формулу

f ' 

f1 

f1 .

(2)

0 2h

Величина называется центральной разностной производной.

Наконец, если взять вторую производную

l'' ( x)  2 f ( x ; x ; x ) 

###  2 

2 0 1 1

f0  f1 

f1  

получаем

( x  x )( x  x

) ( x  x )( x  x

) ( x  x )( x

 x )

 0 1 0 1

1 0 1 1

1 0

1 1 

 f1  2 f0 

h2

f1 ,

приближенную формулу.

f '' 

0

f1  2 f0 

h2

f1 .

Величина f1  2 f0  f1 называется второй разностной производной.

h2

Формулы (1)-(3) называются формулами численного дифференцирования. Погрешности формул (1)-(3) численного дифференцирования.

1. .Предположим, что f C2x0, x1. Тогда существует такая точка , что

f ' 

f1 

f0 

h f ''( ),

x    x .

1. (4)

0 h 2 0 1

1. Если f C3x1, x1, то существует такая точка , что

' f1 

f1

h2 '''

f0 

* f

2h 6

( ),

x1    x1.

1. Когда

f C4x1, x1, то существует такая, что

'' f1  2 f0 

f



0 h2

f1  h2

## 12

f (4)

( ),

x1    x1.

Формулы (4)-(6) называются формулами численного дифференцирования с остаточными членами.

Погрешности формул (1)-(3) оцениваются с помощью следующих неравенств, которые вытекают из соотношений (4)-(6):

f1  f0 h

f  

 h max

f  x ,

0 2 x0 ,x1

f0 

 h2

6

f1  f1

2h

max

x1,x1

f  x ,

f 

 h2

max

f 4  x .

0 12 x1, x1

f1  2 f0  f1 h2

Говорят, что погрешность формулы (1) имеет первый порядок относительно (или порядка

), а погрешность формул (2) и (3) имеет второй порядок относительно (или порядка ). Также говорят, что формула численного дифференцирования (1) первого порядка точности (относительно ), а формулы (2) и (3) имеют второй порядок точности.

Указанным способом можно получать формулы численного дифференцирования для более старших производных и для большего количества узлов интерполирования.

**Определение**. Пусть дан отрезок [a,b]. Равномерной сеткой на этом отрезке назовем множество узлов  h такое, что  h = { xj = jh, j=0,…,n, h=(b-a)/n) }.

**Определение.** Сеточной функцией y = y j = y(xj) называется функция, заданная в узлах сетки.

Любую сеточную функцию y j = y(xj) можно представить в виде вектора Y=(y0, y1, …, yn-1, yn), и, следовательно, множество сеточных функций образует конечномерное пространство, в данном случае размерности n+1. В этом пространстве можно ввести норму, например

Image111 Image112 или .

Image113 Пусть дано дифференциальное уравнение

Lu(x) = f(x,u) ( например, ) .

Заменим Lu в узле сетки xi линейной комбинацией значений сеточной функции yi на некотором множестве узлов сетки, называемом шаблоном. Такая замена Lu на Lh yh называется аппроксимацией на сетке дифференциального оператора L разностным оператором Lh . Замена непрерывной функции f(x,u) в узлах сетки на сеточную функцию  (xh,yh) называется аппроксимацией правой части.

Таким образом дифференциальное уравнение можно аппроксимировать (заменить) на сетке

###### разностной схемой

Image114 Lh yh =  ( xh,yh) ( например, ).

Изучение разностных аппроксимаций проводится сначала локально, т.е. в любом фиксированном узле сетки.

Одна из модификаций метода заключается в том, что сначала вычисляют промежуточные значения

*хi*+1/2 *= хi + h/*2 и *уi*+1/2 *= уi + fi . h/*2

и находят направление поля интегральных кривых в средней точке (*хi*+1/2, *уi*+1/2 ) отрезка [*хi, хi*+1]*,*

т.е.

*fi*+1/2 *= f*(*х i*+1/2*, у i*+1/2)*,*

а затем полагают *уi*+1 = *уi*+1/2.

Пусть uh – проекция непрерывной функции u(x) на сетку ( например, uh = u(xj) = uj ). **Определение.** Погрешностью аппроксимации дифференциального оператора Lu разностным оператором Lh назовем величину  1 = (Lu)h – Lh uh , где (Lu)h – проекция на сетку результата действия дифференциального оператора L на функцию u

Image115 

**Определение.** Говорят, что погрешность аппроксимации дифференциального оператора имеет в узле xi порядок k , если  1(xi) = O(hk) при h  0.

**Определение.** Погрешностью аппроксимации правой части f сеточной функцией  h назовем величину  2 = fh -  h , где fh – проекция на сетку функции f(x,u) (например, f(xj ,uj).

**Определение.** Погрешность аппроксимации правой части имеет в узле xi порядок m , если

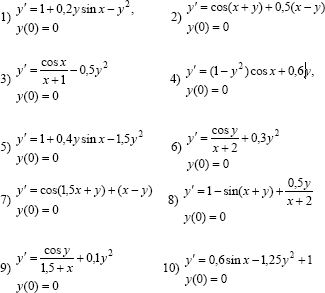
 2 = O(ℎ𝑚𝑚) → 0 при ℎ → 0

###### Задание. Применяя метод Эйлера численно решить задачу Коши для обыкновенного дифференциального уравнения первого порядка.

**Порядок выполнения работы.**

* 1. Изучить метод Эйлера.
  2. Разработать алгоритм решения.
  3. Написать программу для ЭВМ.
  4. Произвести расчеты на ЭВМ.
  5. Оформить отчет: а) теоретическая часть; б) вычислительный алгоритм; в) текст программы; г) результаты.

###### Варианты задания.

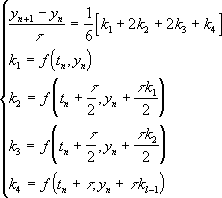


|  |  |
| --- | --- |
| Лабораторное задание №13 | Применение и численная реализация метода Рунге-Кутта четвертого порядка для решения задачи Кощи для обыкновенных дифференциальных уравнений первого порядка. |

Цель работы.

Изучить метод Рунге-Кутта для ОДУ 1-го порядка. Научиться программировать алгоритмы решения.

###### Теоретические сведения. Метод Рунге-Кутта для ОДУ 1-го порядка.

На практике широко распространен Метод Рунге – Кутта четвертого порядка :

Данная схема имеет четвертый порядок аппроксимации.

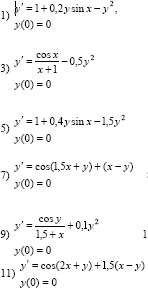
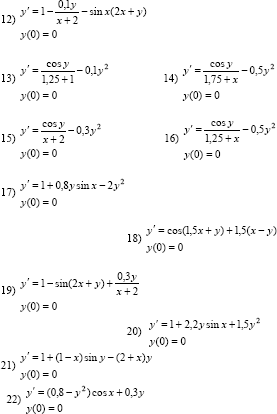
1. **Задание.** Решить приближенно дифференциальное уравнение *y*′ = *f* (*x*, *y*) , удовлетворяющее начальному условию *y*(*x*0 ) = *y*0 на отрезке [0,1] с шагом h=0,1 методом Рунге-Кутта;

Варианты задания приведены.

###### Порядок выполнения работы.

* 1. Изучить метод Рунге-Кутта.
  2. Разработать алгоритм решения.
  3. Написать программу для ЭВМ.
  4. Произвести расчеты на ЭВМ.
  5. Оформить отчет: а) теоретическая часть; б) вычислительный алгоритм; в) текст программы; г) результаты.

###### Варианты задания.



|  |  |  |
| --- | --- | --- |
| Лабораторное занятие №14 | Применение и численная реализация метода Рунге-Кутта четвертого порядка для решения задачи Коши для систем обыкновенных дифференциальных уравнений первого порядка. | 2 |

Цель работы.

Изучить метод Рунге-Кутта для системы ОДУ 1-го порядка. Научиться программировать алгоритмы решения.

###### Теоретические сведения. Метод Рунге-Кутта для системы ОДУ 1-го порядка

Метод Рунге-Кутта позволяет решать системы обыкновенных дифференциальных уравнений (ОДУ) первого порядка.

Рассмотрим систему двух ОДУ 1-го порядка:

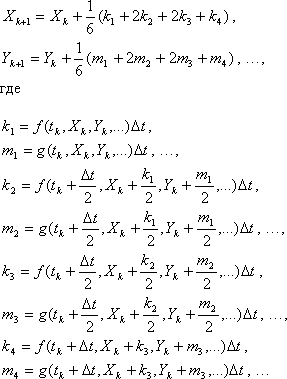
 (1)

которые имеют решение:



где t - независимая переменная (например, время); X, Y и т.д. - искомые функции (зависимые от t переменные). Функции f, g и т.д. - заданы. Также предполагаются **заданными и начальные условия**, т.е. значения искомых функций в начальный момент.

Метод Рунге-Кутта заключается в рекурентном применении следующих формул:



1. **Задание.** Решить приближенно систему дифференциальных уравнений методом Рунге-Кутта с известными начальными условиями;

Варианты задания будут указаны в сборнике задач Н.В.Копченова, И.А.Марон, Вычислительная математика в примерах и задачах.

###### Порядок выполнения работы.

* 1. Изучить метод Рунге-Кутта для системы ОДУ 1-го порядка.
  2. Разработать алгоритм решения.
  3. Написать программу для ЭВМ.
  4. Произвести расчеты на ЭВМ.
  5. Оформить отчет: а) теоретическая часть; б) вычислительный алгоритм; в) текст программы; г) результаты.

|  |  |
| --- | --- |
| Лабораторное занятие №14 | Применение и исследование условия устойчивости метода прогонки для решения краевой задачи для линейного ОДУ второго порядка.  Численная реализация конкретных задач. |

Цель работы.

Изучить метод конечных разностей для решения краевой задачи для линейного обыкновенного дифференциального уравнений второго порядка.. Научиться программировать алгоритмы решения.

###### Теоретические сведения. Метод конечных разностей для решения краевой задачи для линейного обыкновенного дифференциального уравнений второго порядка.

Рассмотрим линейное дифференциальное уравнение второго порядка

*y’’ + р*(*х*)*.у' + q*(*х*)*.у = f*(*х*) (1)

с краевыми условиями:

(2)

где *|*| + || > 0 и *|*| + || > 0, а *р*(*х*)*, q*(*х*)*, f*(*х*) *–* известные функции, непрерывные на отрезке [*а, b*]*.*

Численное решение задачи (1) – (2) состоит в нахождении приближенных значений *у*0*, у*1*,*

*…, уn* искомого решения *у*(*х*) в точках *х*0*, х*1*,* , *…, хn*.

Одним из наиболее распространенных методов решения этой краевой задачи является све- дение ее к системе конечно-разностных уравнений. Используя равномерную сетку, образованную системой равноотстоящих узлов *хi = =х*0 *+ i.h, i* = 0, 1, …, *n,* аппроксимируем *у'*(*х*) и *у»*(*х*) в каждом внутреннем узле центральными разностями

а на концах отрезка – односторонними

Обозначив *х*0 *= а*; *хn = b*; *h =* (*b – а*) */ n*; *р*(*хi*) *= рi*; *q*(*хi*) *= qi*; *f*(*хi*) *= fi*; *у'*(*хi*) *= уi'*; *у»*(*хi*) *= уi»*; *f*(*хi*) *= уi,* получим систему линейных уравнений

(3)

Теперь, чтобы найти приближенные значения *у*0*, у*1*, …, уn* искомого решения, надо решить эту систему из *n+*1 линейных уравнений с *n+*1 неизвестными, что можно сделать любым стан- дартным методом решения линейных систем, который будет называться *конечно-разностным*. Однако матрица последней системы трехдиагональная, поэтому для ее решения применима спе- циальная вычислительная схема, называемая *методом прогонки.*

Перепишем систему (.3) в несколько ином виде:



 *yi* 1

*i*



* *mi*  *yi*
* *ni*  *y*

*i*1

 *h*2  *f* ;



*a*

*y*

 0 0





* *a*1

 *y*1  *y*0  *A*;

*h*

*yn*  *yn*1

(4)

*b*0 *yn*  *b*1  *h*



 *B*,

где *mi =* -2 *+ h.рi ; ni =* 1 *– h.рi + h2.qi* .

Решением уравнения (4) относительно *уi* будет

*yi = fi h2 / mi – ni yi*-1 */ mi – yi*+1 */ mi .* (5)

Предположим, что *уi* уже найдено, тогда (5) можно записать

*уi = сi* (*di – уi*+1)*,* (6)

где надо определить неизвестные *сi* и *di* .

Если *i =* 0*,* то на основании одного из краевых условий (4) имеем

*y*0 = (1 *y*1 – *A h*) / (1 - 0*h*) . (7) Подставим выражение (7) в (4) и, считая *i* = 0, получим

*y*  *f*0 *h*2  *n*0  1 *y*1  *Ah* 

*m*

*m*





1  *y* ,

откуда выразим явно *у*1

*m*

2

1

0 0 1

.

  0 *h* 0

Сравнив последнее равенство с равенством (6), определим *с*0 и *d*0

 *a*1 *y*1  *Ah*

*c*0  *m*

 *a*

* *a h* *n a* ;

 0 1 0 0 0 (8)



*d*  *n*0 *Ah*  *f h*2 .

 0

0

*a*1  *a*0 *h*

Теперь последовательно будем искать *сi* и *di.* Для этого выразим *уi-*1 из равенства (6) и подставим полученное выражение в выражение (5), из которого после элементарных преобразований найдем *уi* :

.

Сравним последнее выражение с равенcтвом (4.7), определим *сi* и *di* :

 1

*c*



 *i m*  *n c* ;

(9)

 *i i i*1

*d*  *f h*2  *n c d* .

 *i i*

*i i*1

*i*1

И, наконец, при *i = n* используем второе краевое условие из системы (4), подставив в него

*уn-*1 из равенства (6):

*y*  1*cn*1*dn*1  *B*  *h* . (10)

*n*  1  *c*



1

*n*1

   0  *h*

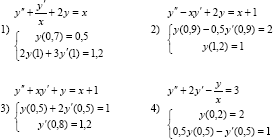
1. **Задание.** Решить краевую задачу для обыкновенного дифференциального уравнения второго порядка с точностью 0,001 и шагом h= 0,1.

Варианты задания приведены.

###### Порядок выполнения работы.

* 1. Изучить метод конечных разностей для решения краевой задачи для линейного обыкновенного дифференциального уравнений второго порядка.
  2. Разработать алгоритм решения.
  3. Написать программу для ЭВМ.
  4. Произвести расчеты на ЭВМ.
  5. Оформить отчет: а) теоретическая часть; б) вычислительный алгоритм; в) текст программы; г) результаты.

###### Варианты задания.



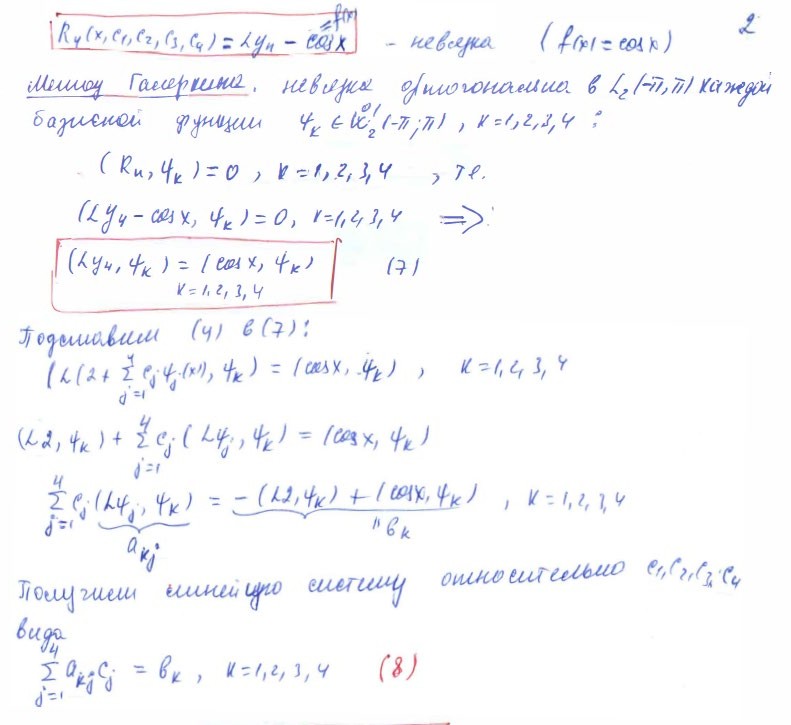
|  |  |
| --- | --- |
| P1053C1T19#yIS1 | P1055C2T19#yIS1 |

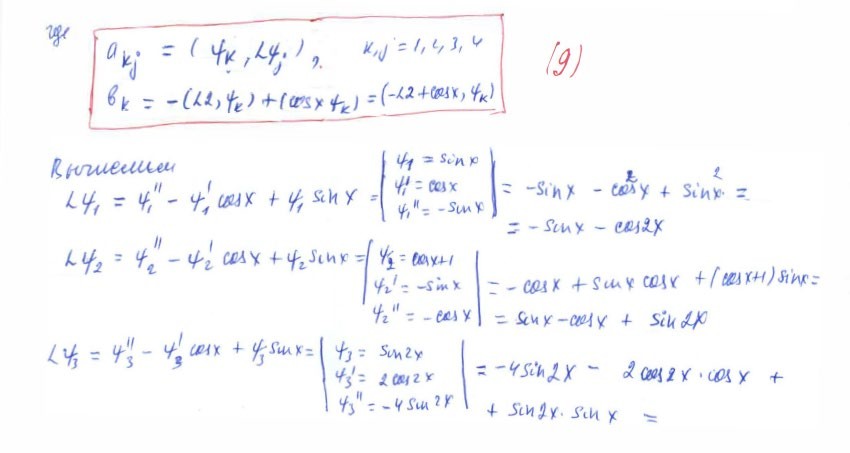
|  |  |
| --- | --- |
| Лабораторное  занятие №15 | Численная реализация краевых задач для линейного ОДУ второго  порядка методом Галеркина |

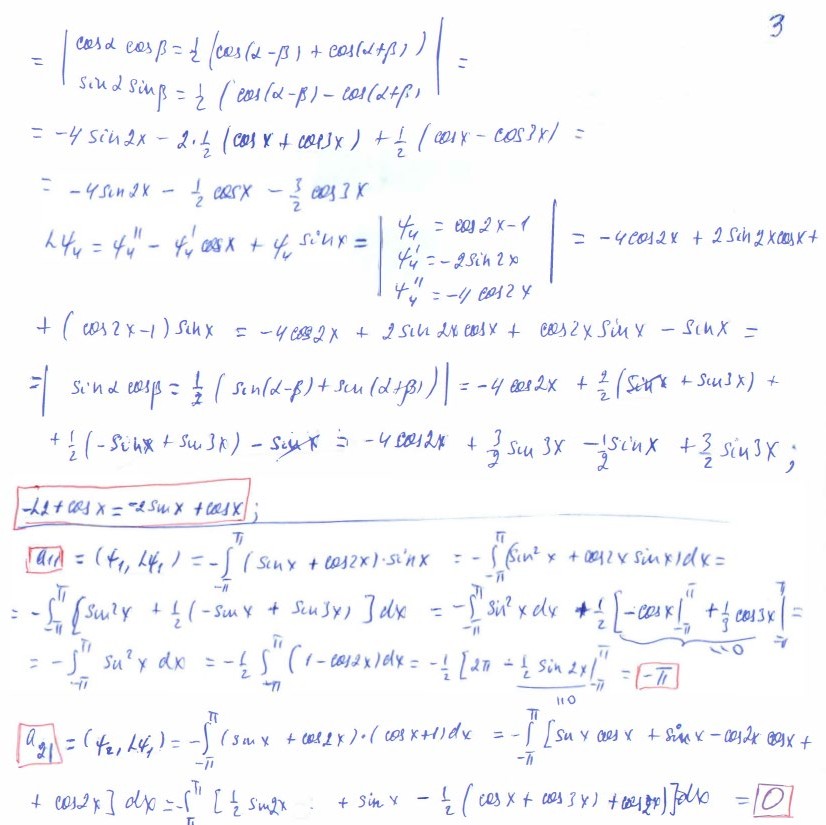
Цель работы.

Изучить метод Галеркина для решения краевой задачи для линейного обыкновенного дифференциального уравнений второго порядка. Научиться составлять линейную систему, применять численные метолы для решения полученной линейной системы.

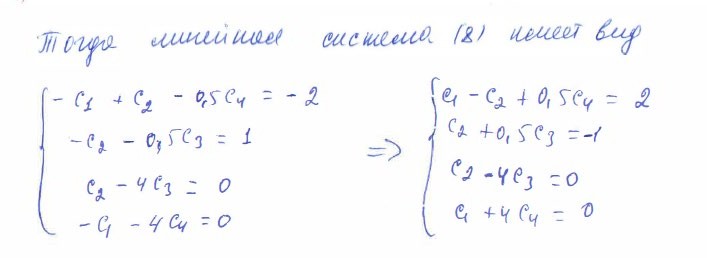
###### Указание для выполнения работы на конкретном пример.







И т.д

𝑎22=−𝜋, 𝑎32=𝜋, 𝑎42=0, 𝑎13=0, … , 𝑎44=−𝜋, 𝑏1=−2𝜋, 𝑏2=𝜋, 𝑏3=𝑏4=0

Находим решение этой системы

с =80, с = 8, с = − 2, с = − 20,

1 63 2 9 3 9 3 63

Тогда приближенное решение имеет вил

10 80 8 2 20

𝑦3(𝑥) =

+ 𝑠𝑖𝑖𝑛𝑥 − 𝑐𝑜𝑠𝑥 − 𝑠𝑖𝑖𝑛2𝑥 − 𝑐𝑜𝑠2𝑥 7 63 9 9 63