##### ЛЕКЦИИ

**Инновационные методы исследования природно-техногенных систем**

**Лекции**

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

Некоторые сведения погрешности. Норма матрицы.

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

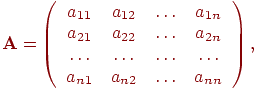
� = 𝛿𝛿(𝑎) + 𝛿𝛿(𝑏)

Рассматриваются наиболее употребительные приближенные методы решения систем линейных алгебраических уравнений. Вводятся согласованные нормы векторов и матриц. Вычисляется число обусловленности в различных нормах.

Постановка задачи Рассмотрим СЛАУ вида

$ \mathbf{Au}= \mathbf{f},$ 

где — невырожденная ( ) квадратная матрица размером n × n



(1.1)

— вектор-столбец решения, — вектор-столбец правой части.



Так как матрица системы — невырожденная, , то решение системы (2.1) существует и единственно.

Из курса линейной алгебры известно правило Крамера нахождения решения. Так, каждый компонент вектора неизвестных может быть вычислен как

$ u_i  = \frac{\Delta_i}{\Delta }, $ 

где Δi — определитель матрицы, получаемой из \mathbf{A} заменой i столбца столбцом правых частей. Время расчета для n = 100 на существующих в момент написания книги компьютерах будет измеряться годами.

На самом деле в настоящее время с помощью компьютеров численно решаются СЛАУ намного более высокого порядка (примерно до n ≈ 106). Такие решения осуществляются при помощи прямых или итерационных численных методов. Прямые методы позволяют в предположении отсутствия ошибок округления (при проведении расчетов на идеальном, т.е. бесконечноразрядном компьютере) получить точное решение задачи за конечное число арифметических действий. Итерационные методы, или методы последовательных приближений, позволяют вычислить последовательность \{{\mathbf{u}}_k\} , сходящуюся к решению задач при k \to \infty (на практике, разумеется, ограничиваются конечным k, в зависимости от требуемой точности).

Однако неточность в задании правых частей и элементов матрицы \mathbf{A} может приводить к значительным погрешностям при вычислении решения (1.1).

Согласованные нормы векторов и матриц

В векторном n-мерном линейном нормированном пространстве введем следующие нормы вектора: кубическая:

{\|\mathbf{u}\|}_1  = \max\limits_{1 \le i \le n}|u_i|, 

октаэдрическая:

{\|\mathbf{u}\|}_2  = \sum\limits_{i = 1}^n|u_i|, 

евклидова (в комплексном случае — эрмитова):

(1.2а)

(1.2б)

(1.2в)



Рассмотрим квадратную матрицу и связанное с ней линейное преобразование \mathbf{v}= \mathbf{Au} , где

(Ln — n-мерное линейное нормированное пространство). Норма матрицы определяется как действительное неотрицательное число, характеризующее это преобразование и определяющееся как

$ \|\mathbf{A}\|= \sup\limits_{\|\mathbf{u}\| \ne 0} \frac{\|\mathbf{Au}\|}{\|\mathbf{u}\|}$ (1.3)

Укажем некоторые свойства нормы матрицы:



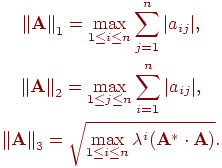
Заметим, что норму матрицы (1.3) называют подчиненной норме вектора. Говорят, что норма матрицы согласована с нормой вектора \mathbf{u} , если выполнено условие



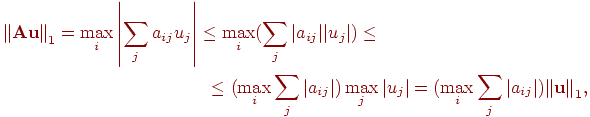
Нетрудно видеть, что подчиненная норма согласована с соответствующей метрикой векторного пространства. В самом деле



Согласованные с введенными выше нормами векторов нормы матриц будут определяться следующим образом:



{\|\mathbf{A}\|}_1  Покажем, как получается выражение для согласованной нормы матрицы , соответствующей норме вектора {\|\mathbf{u}\|}_1 .



Вычислим норму вектора

:

откуда

\frac{{\|\mathbf{Au}\|}_1}{{\|\mathbf{u}\|}_1}\le \max\limits_i \sum\limits_j |{a_{ij}}|. 

По определению нормы матрицы как точной верхней грани отношения

\frac{\|\mathbf{Au}\|}{\|\mathbf{u}\|}, \max\limits_j \sum\limits_i{|{a_{ij}}|} = {\|\mathbf{A}\|}_1 , 

если существует вектор, на котором точная верхняя грань достигается. Покажем, что таким вектором является, например,

{\mathbf{v}}_k = \{sign a_{k1}, \ldots , sign a_{kn}\}^T, 

\|{\mathbf{v}}_k\|= k при этом допустим, что максимум в последнем неравенстве достигается при i = k. Поскольку , то \sum\limits_j{a_{kj}v_j} = \sum\limits_j {|a_{kj}|} =  \max\limits_i \sum\limits_j{|{a_{ij}}|} .

Тогда, в соответствии с выражением для первой нормы вектора, получаем

\|{\mathbf{Av}}\|}_1 = \max\limits_i \sum\limits_j{|{a_{ij}}|}. 

{\|\mathbf{A}\|}_1  = \max \sum\limits_j{|{a_{ij}}|} Таким образом, точная верхняя грань в рассмотренном неравенстве достижима и действительно

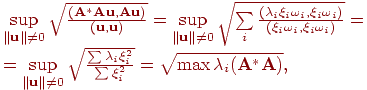
.

Для третьей нормы (2.2в)



Заметим, что матрица — симметричная. Без ограничения общности предположим, что все собственные числа матрицы различны. Матрица обладает всеми действительными собственными значениями, и каждому собственному числу соответствует собственный вектор. Все собственные векторы взаимно ортогональны. Можно рассмотреть ортонормированную систему собственных векторов ω1, … , ωn; λ1, … , λn — соответствующие им собственные значения. Любой вектор \mathbf{u}  можно представить в виде своего разложения по базису из собственных векторов: \sum\limits_i{\xi_i\omega_i} . Кроме того, (\mathbf{A}^*\mathbf{A})\omega_i  = \lambda_i \omega_i . Поэтому

причем точная верхняя грань достигается при . Действительно,



т.к. откуда

В важном частном случае симметричной (самосопряженной) матрицы \mathbf{A} имеем

\lambda_{\mathbf{A}^* \mathbf{A}}^i  = \lambda_{{\mathbf{A}}^2}^i  = {|{\lambda_{\mathbf{A}}^i}|}^2, поэтому {\|\mathbf{A}\|}_3  = \max\limits_i |{\lambda_{\mathbf{A}}^i}|. 

*Обусловленность СЛАУ. Число обусловленности матрицы*

Понятия согласованных норм матриц и векторов позволяют оценить погрешности, возникающие при численном решении СЛАУ. Пусть и матрица, и правая часть системы заданы с некоторой погрешностью, тогда наряду с системой

\mathbf{Au}= \mathbf{f} 

рассматривается система

(\mathbf{A}+ \Delta\mathbf{A})(\mathbf{u}+ \Delta\mathbf{u}) = \mathbf{f}+ \Delta\mathbf{f}. 

Теорема. Пусть правая часть и невырожденная матрица СЛАУ (2.4) вида

(1.4)

(1.5)

получили приращения \Delta\mathbf{f} и \Delta\mathbf{A} соответственно. Пусть существует обратная матрица и выполнены условия

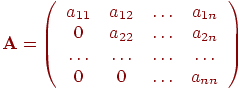


где . В этом случае оценка относительной погрешности решения \|{\Delta\mathbf{u}}\|/\|\mathbf{u}\|  удовлетворяет неравенству

$  \frac{\|\Delta\mathbf{u}\|}{\|\mathbf{u}\|} \le \frac{\mu} {1 - \mu \frac{\|\Delta\mathbf{A}\|}{\|\mathbf{A}\|}} \left({\frac{\|\Delta \mathbf{f}\|}{\|\mathbf{f}\|}+ \frac{\|\Delta\mathbf{A}\|} {\|\mathbf{A}\|}}\right). $ 

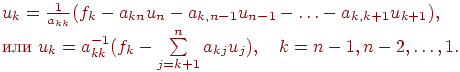
**Прямые методы решения СЛАУ**

Трудность численного решения рассматриваемых СЛАУ определяется видом матрицы \mathbf{A} . Легко получается решение системы с диагональной матрицей, в этом случае система распадается на n линейных уравнений, каждое из которых содержит лишь одну неизвестную величину. Для диагональной системы очевидны явные формулы. В случае треугольной матрицы



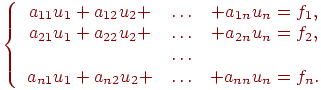
из последнего уравнения получаем .

Решая систему линейных уравнений с треугольной матрицей «снизу вверх», для uk имеем

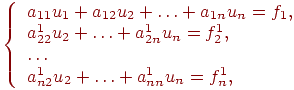


Можно оценить количество арифметических действий, затрачиваемых на решение такой системы. Оно составляет O(n2) .

Пусть теперь система уравнений имеет матрицу общего вида. Стандартная схема такого решения разделяется на два этапа: прямой ход — приведение матрицы к треугольному виду, и обратный — вычисление решения системы. **Метод исключения Гаусса**

Рассматривается система уравнений

(1.10)

Прямой ход метода Гаусса состоит в следующем. Положим, что a_{11} \ne 0 и исключим u1 из всех уравнений, начиная со второго, для чего ко второму уравнению прибавим первое, умноженное на –a21/a11 = - η21, к третьему прибавим первое, умноженное на –a31/a11 = - η31 и т.д. После этих преобразований получим эквивалентную систему:

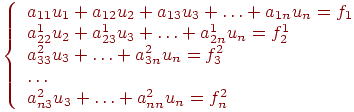
(1.11)

в которой коэффициенты и правые части определяются следующим образом:

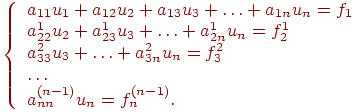


Теперь положим . Аналогично, вычислив множители второго шага

исключаем u2 из последних (n – 2) уравнений системы (2.17). В результате преобразований получим новую эквивалентную систему уравнений

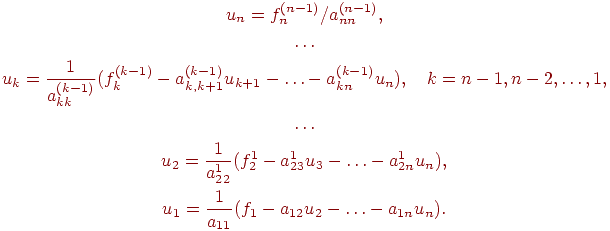


в которой . Продолжая алгоритм, т.е.

исключая ui (i = k + 1, …, n), приходим на n – 1 шаге к системе с треугольной матрицей

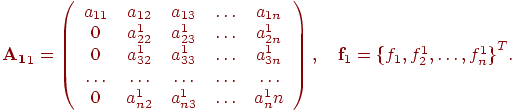
(1.12)

Обратный ход метода Гаусса позволяет определить решение системы линейных уравнений. Из последнего уравнения системы находим un; подставляем это значение в предпоследнее уравнение, получим un-1. Поступая так и далее, последовательно находим un-2, un-3, …, u1. Вычисления компонент вектора решения проводятся по формулам

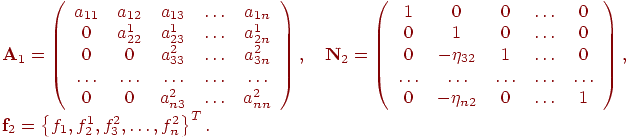
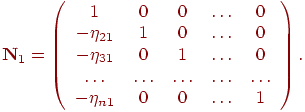


a_{11} \ne 0 a_{22} \ne 0 Этот алгоритм прост и легко реализуем при условии, что , и т.д. Количество арифметических действий прямого хода ≈ 2/3n3, обратного ≈ n2. Это уже приемлемая для современных компьютеров величина.

Рассмотрим метод Гаусса с позиции операций с матрицами. Пусть {\mathbf{A}}_1 — матрица системы после исключения первого неизвестного



{\mathbf{A}}_2\mathbf{u}={\mathbf{f}}_2 Введем новую матрицу



Очевидно,

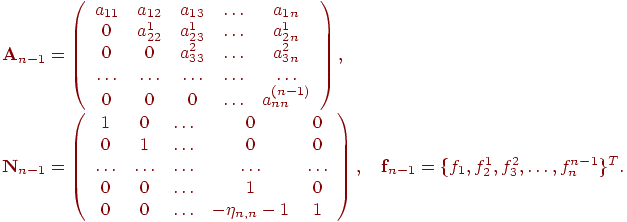
, где

,

. Аналогично, после второго шага система приводится к виду

,

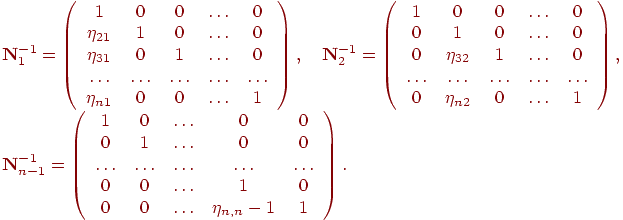
После n – 1 шага получим {\mathbf{A}}_{n - 1}\mathbf{u} = {\mathbf{f}}_{n - 1}, {\mathbf{A}}_{n - 1} = {\mathbf{N}}_{n - 1} \cdot {\mathbf{A}}_{n - 2}, {\mathbf{f}}_{n - 1} = {\mathbf{N}}_{n - 1}{\mathbf{f}}_{n - 2}, 



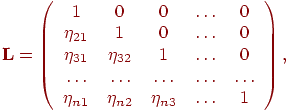
{\mathbf{f}}_{(n - 1)} = {\mathbf{N}}_{n - 1} \ldots {\mathbf{N}}_2{\mathbf{N}}_1\mathbf{f}, В итоге получаются матрица и вектор



откуда . При этом



\mathbf{L} = {\mathbf{N}}_1^{- 1}{\mathbf{N}}_2^{- 1} \ldots  {\mathbf{N}}_{n - 1}^{- 1} После введения обозначений \mathbf{U} = {\mathbf{A}}_{n - 1} , , где



получим \mathbf{A} = \mathbf{LU} .

Это представление матрицы называется ***LU-разложением*** (на произведение нижней и верхней треугольных матриц и ). Прямой ход *метода Гаусса* можно рассматривать как один из вариантов представления матрицы в виде произведения двух треугольных матриц, или *LU- разложения*. Его можно провести и другими способами.



В реальных вычислениях используются методы с *выбором главного* (или **ведущего**) **элемента.** *Выбор главного элемента* **по столбцам** реализуется следующим образом: перед исключением u1 отыскивается \max\limits_i|{a_{i1}}| . 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.

\max\limits_j|{a_{kj}}| Затем отыскивается \max\limits_i|{a_{i2}^1}| , и процедура поиска *главного элемента* в столбцах повторяется. Так же реализуется *выбор главного элемента* по строкам: перед исключением u1 отыскивается

. 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**

|{a_{ii}}| \ge \sum\limits_{\substack{j = 1 \\  j \ne i}}^n{|{a_{ij}}|} 

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.**

{\mathbf{r}}^1 = \mathbf{f} - {\mathbf{Au}}^1 {\mathbf{u}}^1 {\mathbf{\varepsilon}}^1 = \mathbf{u} - {\mathbf{u}}^1 {\mathbf{A\varepsilon}}^1 = {\mathbf{r}}^1 {\mathbf{A\varepsilon}}^1 = \mathbf{Au} - {\mathbf{Au}}^1 = \mathbf{f}- {\mathbf{Au}}^1 Полученное решение можно улучшить следующим образом. Пусть есть невязка, допущенная при решении рассматриваемой системы ( — полученное численное решение) за счет ошибки округлений. Очевидно, что погрешность удовлетворяет *СЛАУ*

, так как .

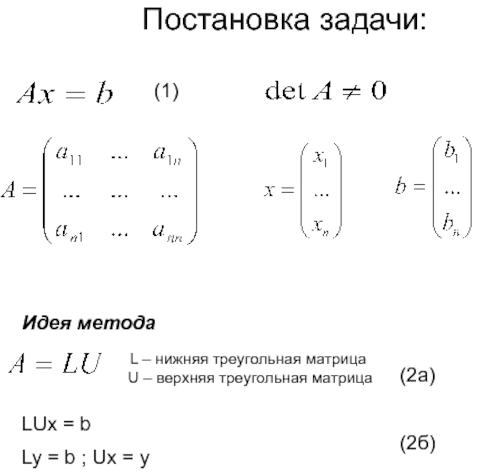
Решив последнюю систему, получаем \mathbf{\varepsilon}^1 , после чего уточняем решение:

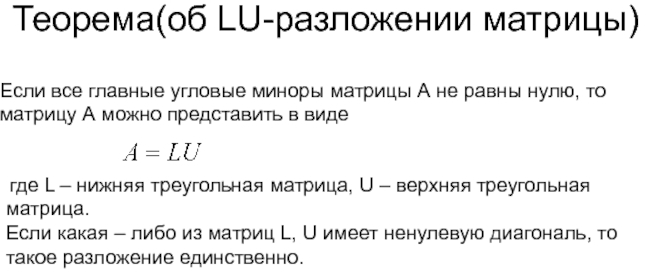
{\mathbf{u}}^2 = {\mathbf{u}}^1 + {\mathbf{\varepsilon }}^1. 

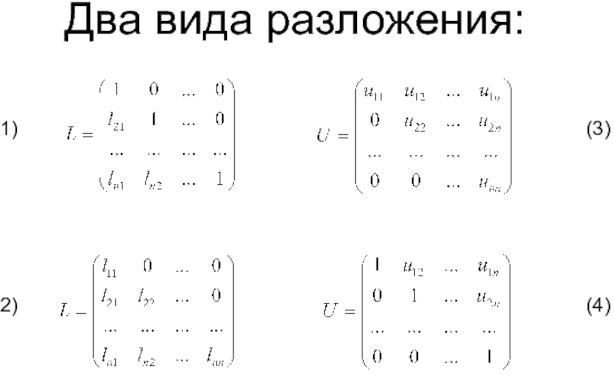
Эту процедуру можно продолжить.

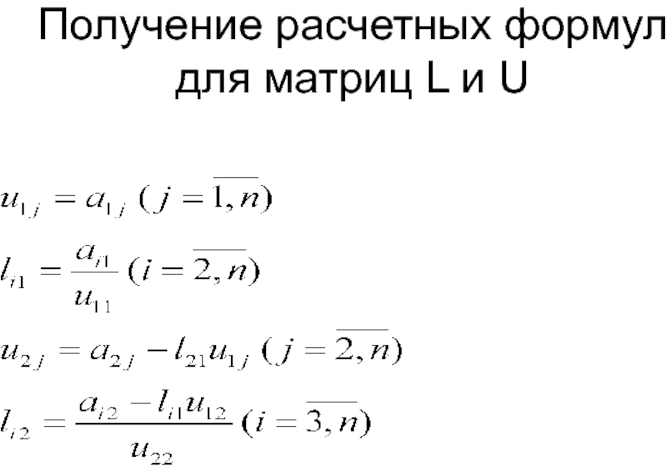
|  |  |  |
| --- | --- | --- |
| **Лекционное занятие №2** | Прямые методы решения СЛАУ: метод LU разложения, метод Холецкого (метод квадратного корня) | **1** |

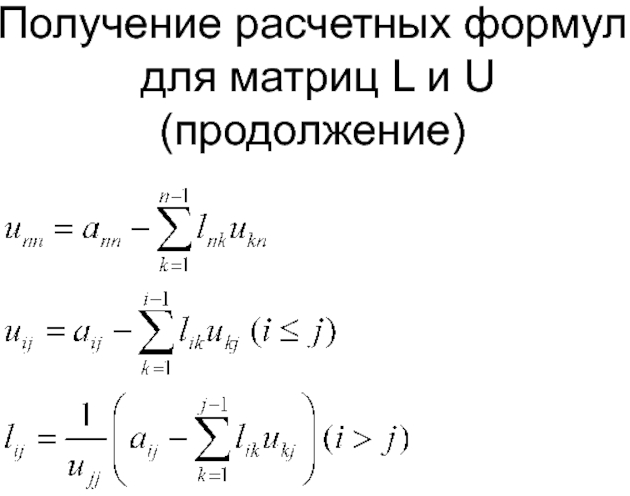
##### Метод LU разложения для решения СЛАУ

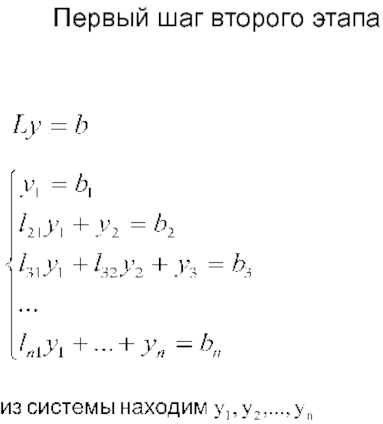


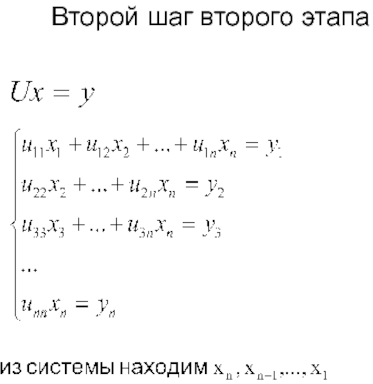












1. **Метод квадратного корня для решения линейных систем**

*Краткая характеристика метода.*

Метод квадратного корня применяется в том случае, когда матрица А симметричная, то есть:

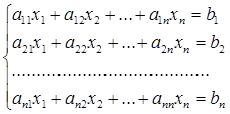
aij = aji (i, j = 1, 2, …, n).

Кроме того, матрица должна быть невырожденной, то есть её определитель не должен равняться нулю (det(A)0). Таким образом, система будет иметь единственное решение.

Метод квадратного корня дает большой выигрыш во времени по сравнению с другими методами (например, методом Гаусса), так как, во-первых, существенно уменьшает число умножений и делений (почти в два раза для больших n).

*Постановка задачи*

К решению систем линейных уравнений сводятся многочисленные практические задачи. Рассмотрим систему n линейных алгебраических уравнений с n неизвестными:



Систему уравнений (1) можно записать в векторно-матричном виде:

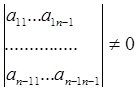
Ax=b (1)

Таким образом, задача состоит в том, чтобы вычислить столбец неизвестных, используя метод квадратного корня.

Пусть дана симметричная система линейных уравнений в матричном виде (1): Ах=b

К ee решению может быть применена идея разложения матрицы А в произведение двух матриц специального вида. Основанием для этого служит следующая теорема [Крылов В.И. и др. Вычислительные методы, т.I. М.: Наука, 1976.]:

Теорема. Какова бы ни была матрица А с отличными от нуля главными минорами:





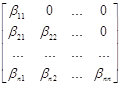
, … ,

ее всегда можно разложить в произведение двух треугольных матриц:

A=BC,

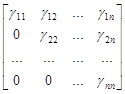
где В - левая треугольная матрица:

В=



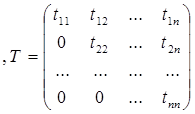
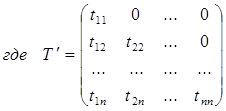
С - правая треугольная матрица:

С=

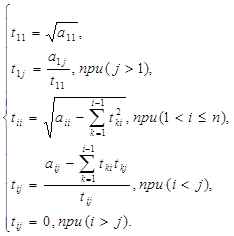


Так как данная матрица (1) симметрична, то она раскладывается на произведение двух взаимно транспонированных треугольных матриц [Вержбицкий В.М. Основы численных методов. М.: Высшая школа. 2002]:

А = Т Т, (2)



Получим следующие формулы для определения tij [Крылов В.И. и др. Вычислительные методы, т.I. М.: Наука, 1976]:



Далее, решение системы сводится к решению двух треугольных систем. Действительно, равенство (1) равносильно двум равенствам:

ľ'y=b и ľx=y. (3)

**Список используемой литературы**

1. Бахвалов Н.С., Жидков Н.П., Кобельков Г.М. Численные методы. М.: Наука, 2001.
2. Вержбицкий В.М. Основы численных методов. М.: Высшая школа. 2002.
3. Крылов В.И. и др. Вычислительные методы, т.I. М.: Наука, 1976.
4. Трубников С.В Численные методы. Часть 1: Теория погрешностей. Решение алгебраических и трансцендентных уравнений и систем: Учебное пособие для студентов вузов. - Брянск: Изд-во БГУ, 2005.
5. Фаддеев Д.К., Фаддеева В.Н. Вычислительные методы линейной алгебры. М., Л.: изд-во физ.- мат. лит-ры, 1963.
6. Лапчик М.П., рагулина М.И., Хернер Е.К. Численные методы.- М.: Наука, 2007.
7. Заварыкин В.М., Житомирский Г.В. Численные методы.- М.: Просвещение 1990.
8. Березин И.С., Жидков Н.П. Методы вычислений, Т. 1, 2. М.: Наука, 1966.
9. Воеводин В.В., Кузнецов Ю.А., Матрицы и вычисления.- М.: Наука, 2007.

|  |  |  |
| --- | --- | --- |
| **Лекционное занятие №3** | Прямые методы решения СЛАУ. Метод прогонки для СЛАУ с  трехдиагональной матрицей. Достаточное условие применимости метода прогонки. | **1** |

Метод прогонки применяется для решения систем уравнений с трехдиагональной (ленточной) матрицей. Является частным случаем метода Гаусса и состоит из прямого и обратного хода. Прямой ход состоит в исключении элементов матрицы системы, лежащих ниже главной диагонали.

**Метод прогонки** или **алгоритм Томаса** (англ. *Thomas algorithm*) используется для решения систем линейных уравнений вида ~Ax=F , где *A* — трёхдиагональная матрица.

***Описание метода.***

Система уравнений ~Ax=F  равносильна соотношению

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

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

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

Используя это соотношение, выразим xi-1 и xi через xi+1 и подставим в уравнение (1):

\left(A_i\alpha_i\alpha_{i+1} + C_i\alpha_{i+1} + B_i\right)x_{i+1} + A_i\alpha_i\beta_{i+1} + A_i\beta_i + C_i\beta_{i+1} - F_i = 0 ,

где *Fi* — правая часть *i*-го уравнения. Это соотношение будет выполняться независимо от решения, если потребовать

\begin{cases} A_i\alpha_i\alpha_{i+1} + C_i\alpha_{i+1} + B_i = 0\\  A_i\alpha_i\beta_{i+1} + A_i\beta_i + C_i\beta_{i+1} - F_i = 0 \end{cases} 

Отсюда следует:

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} 

Из первого уравнения получим:

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

После нахождения прогоночных коэффициентов α и β, используя уравнение (2), получим решение системы.

Обратная прогонка

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



|  |  |  |
| --- | --- | --- |
| **Лекционное занятие №4** | Итерационные методы решения СЛАУ. Метод простой итераций (МПИ), достаточное условие сходимости МПИ, оценки погрешности. Методы Якоби и Зейделя, релаксации, достаточные условия их сходимости. | **1** |

**Канонический вид одношаговых итерационных методов** для решения системы линейных алгебраических уравнений A**x** = **f** :



где Вк+1 – квадратная матрица n× n ,  k+1 > 0 – итерационный параметр. В дальнейшем будем использовать следующие согласованные нормы в конечномерном пространстве размерности n:

* евклидова норма
* норма в С
* энергетическая норма A=A\*>0

Под нормой матрицы А будем понимать Итерационный метод сходится, если .

**Определение.** Величина zk = xk – x называется погрешностью решения.

**Определение**. Если Bk+1 = B и  k+1 =  то метод называется стационарным

Теорема.

Пусть A=A\* >0, и B- 0.5 A>0 тогда итерационный метод



сходится в норме   \*   A , .

Метод Зейделя

Каноническому виду метода Зейделя соответствует B=(D+L),  =1, где D –диагональная матрица, L – нижняя треугольная матрица

Индексный вид метода Зейделя



Теорема

Пусть A=A\*>0, тогда метод Зейделя сходится*.*

Теорема

Пусть матрица A такова, что i=1,…, n,  q <1, тогда метод Зейделя сходится со скоростью геометрической прогрессии со знаменателем q, т.е.

.

Метод релаксации

Канонический вид



 < 1- метод нижней релаксации

 = 1- метод Зейделя

 > 1- метод верхней релаксации

**Теорема** Пусть A=A\*>0, 0< <2, тогда метод релаксации сходится.

Метод простой итерации

Канонический вид метода простой итерации –



Теорема

Пусть A=A\*>0, , тогда метод простой итерации сходится.

**Замечание.** Если A=A\*>0 то у матрицы А существует n собственных значений  k таких, что 0<  min= 1< 2<…< n-1< n= max,

при этом и .

Можно найти оптимальное значение итерационного параметра  =  0 такое, что заданная точность решения будет достигаться за минимальное число итераций.

Теорема

Пусть A=A\*>0, тогда при для погрешности явного метода простой итерации zk справедлива оценка

.

|  |  |  |
| --- | --- | --- |
| **Лекционное занятие №5** | Итерационные методы вариационного типа. Метод минимальных невязок, метод наискорейшего спуска | **1** |

Связь между вариационной задачей и задачей решения СЛАУ

Пусть \mathbf{u} \in L^n   где Ln есть n-мерное евклидово пространство. Рассмотрим квадратичный



,

функционал от , называемый **функционалом энергии**:

\Phi (\mathbf{u}) = (\mathbf{Au,u}) - 2(\mathbf{f,u}) + c,  

(\mathbf{u},\mathbf{A}^*\mathbf{u}) = (\mathbf{A}^*\mathbf{u},\mathbf{u}) где \mathbf{A} — линейный оператор, \mathbf{f} \in L^n , c — константа. Этот функционал совпадает с квадратичным функционалом где \mathbf{A}^* — сопряженный к \mathbf{A}  оператор. Действительно, по определению сопряженного оператора и



в силу коммутативности скалярного произведения. Тогда



Без ограничения общности предположим, что оператор \mathbf{A} — самосопряженный, \mathbf{A} = \mathbf{A}^* . В противном случае будем рассматривать задачу с оператором

\frac{1}{2}(\mathbf{A}+{\mathbf{A}^*})  

при решении вариационной задачи.

\mathbf{u} Будем также считать, что \mathbf{A} — положительный оператор, т.е. \mathbf{A} > 0 , это означает, что для любого ненулевого вектора выполнено {(\mathbf{Au}, \mathbf{u}) > 0} .

\Phi (\mathbf{u}) Поставим задачу об отыскании элемента \mathbf{v} , придающего наименьшее значение функционалу

:

\Phi ({\mathbf{v}}) = \min\limits_{\mathbf{u} \in L^n}\Phi (\mathbf{u}).  

**Теорема.** Пусть \mathbf{A} = {\mathbf{A}*} > 0 . В этом случае существует единственный элемент {\mathbf{v}} \in L^n  ,

\Phi (\mathbf{u}) =  (\mathbf{Au,u}) - (2\mathbf{f,u}) + c, придающий наименьшее значение квадратичному функционалу

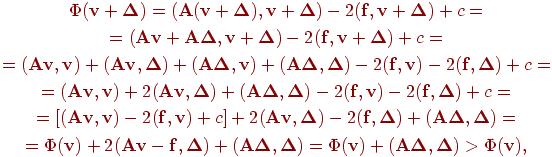
являющийся решением *СЛАУ \mathbf{Au}= \mathbf{f}. *

Доказательство.

\mathbf{v} {\mathbf{Av}} - \mathbf{f} = 0 \mathbf{u} = \mathbf{v} *СЛАУ \mathbf{Au}= \mathbf{f} *имеет единственное решение , поскольку \mathbf{A} является невырожденным оператором в силу его положительной определенности. Покажем, что в этом случае при для любого вектора Δ имеет место \Phi ({\mathbf{v}} + {\mathbf{\Delta }}) > \Phi ({\mathbf{v}}), т.е. при достигается

минимум квадратичного функционала \Phi (\mathbf{u}) .

Действительно,



\mathbf{Av} = \mathbf{f} т.е. при и любом Δ имеет место \min\limits_\mathbf{u}\Phi (\mathbf{u}) . Докажем, что верно и обратное утверждение. Если элемент доставляет минимальное значение функционалу энергии, то он является решением системы линейных уравнений \mathbf{Av} = \mathbf{f} . Из курса математического анализа известно, что в точке



минимума должно выполняться условие . Вычисляя градиент,

\Phi (\mathbf{u}) приходим к условию минимума функционала . Таким образом установлена эквивалентность вариационной задачи (отыскание элемента, придающего минимум

) и задачи о нахождении решения *СЛАУ*.

Заметим, что *СЛАУ* с самосопряженным и положительно определенным оператором \mathbf{A}  представляют собой важный класс задач в математической физике, в частности, они возникают при решении краевых задач для эллиптических уравнений. При необходимости можно произвести симметризацию по Гауссу исходной системы.

Методы наискорейшего спуска

Метод градиентного спуска состоит в нахождении следующего приближения в итерационном процессе из предыдущего, путем смещения в направлении градиента функционала



|  |
| --- |
| (1) |
| (2) |

{\alpha}_k где \mathbf{A} — положительно определенная симметричная матрица; — параметр, определяемый из заданных условий; например, из условия минимума величины

\Phi \left[{{\mathbf{u}}_k - \alpha_k \cdot grad \Phi ({\mathbf{u}}_k)}\right].  

grad \Phi (\mathbf{u}) = 2(\mathbf{Au}- \mathbf{f}), В этом случае *итерационный метод* называется ***методом наискорейшего спуска***. Так как то (2) приобретает вид

 (3)

\Phi (\tau_k, {\mathbf{u}}_{k + 1}) что соответствует записи *итерационного метода*. Здесь τk является итерационным параметром, который в *методе наискорейшего спуска* определяется из условия минимума функции

по τk. Найдем условие этого минимума:

0 = 2(\mathbf{Au}_{k + 1} - \mathbf{f}, {({\mathbf{u}}_{k + 1})^{\prime}}_{\tau_k}) =    - 2(\mathbf{Au}_{k + 1}- \mathbf{f}, \mathbf{Au}_k  - \mathbf{f}).  

({\mathbf{Au^{\prime}}},\mathbf{u}) =  ({\mathbf{u^{\prime}}},\mathbf{A}^*\mathbf{u}) =  (\mathbf{Au}{\mathbf{,u^{\prime}}}), (\mathbf{v,Aw}) = (\mathbf{Aw,v}) Здесь учтено соотношение: поскольку \mathbf{A} =  \mathbf{A}^* и в силу самосопряженности оператора . Подставим в последние



равенства {\mathbf{u}}_{k+1} из (3), получим откуда следует

\mathbf{Au}_k - \mathbf{f},\mathbf{Au}_k - \mathbf{f}) - \tau_k(\mathbf{A}(\mathbf{Au}_k - \mathbf{f}), \mathbf{Au}_k - \mathbf{f}) = 0,  

\tau_k = (\mathbf{Au}_k - \mathbf{f},\mathbf{Au}_k -  \mathbf{f})/(\mathbf{A}(\mathbf{Au}_k - \mathbf{f}),\mathbf{Au}_k - \mathbf{f}) , или

.



где

Вектор {\mathbf{r}}_k называют **вектором невязки.** **Метод минимальных невязок**

Этот *итерационный метод* определяется следующим образом. Пусть {\mathbf{u}}_{k+ 1} = {\mathbf{u}}_k - \tau_k{\mathbf{r}}_k, как и ранее, {\mathbf{r}}_k = \mathbf{Au}_k - \mathbf{f} . Итерационный параметр τk на каждой итерации выбирается так, чтобы

минимизировать евклидову норму невязки {\mathbf{r}_{k+1}} . Заметим, что итерационный процесс может быть представлен в равносильном виде в терминах невязки



. Тогда для квадрата евклидовой (третьей) нормы невязки получаем

условие

({\mathbf{r}}_{{k+ 1}},{\mathbf{r}}_{{k+ 1}}) = ({\mathbf{r}}_k,{\mathbf{r}}_k) - 2\tau_k({\mathbf{Ar}}_k,{\mathbf{r}}_k) + \tau_k^2 ({\mathbf{Ar}}_k,{\mathbf{Ar}}_k).  

Для отыскания минимума невязки на следующей итерации приравняем нулю производную последнего выражения по итерационному параметру τk. Получим равенство

 - 2({\mathbf{Ar}}_k,{\mathbf{r}}_k) +   2\tau_k({\mathbf{Ar}}_k,{\mathbf{Ar}}_k) = 0.  

Из последнего соотношения находим значение итерационного параметра

$ \tau_k= \frac{({\mathbf{Ar}}_k,{\mathbf{r}}_k)}{({\mathbf{Ar}}_k,{\mathbf{Ar}}_k)}. $ 

|  |  |  |
| --- | --- | --- |
| **Лекционное занятие №6** | Метод простой итерации (МПИ) и его сходимость для решения нелинейных уравнений и систем нелинейных уравнений | **1** |

Нелинейные уравнения можно разделить на два класса: **алгебраические и трансцендентные.** Алгебраическими называют уравнения, содержащие только алгебраические функции (целые, рациональные, иррациональные). В частности, многочлен является целой алгебраической функцией. Уравнения, содержащие другие функции (тригонометрические, показательные, логарифмические и др.) называются трансцендентными. Методы решения нелинейных уравнений делятся на две группы: точные и итерационные. Точные методы позволяют записать корни в виде некоторого конечного соотношения (формулы).

Как известно, многие уравнения и системы уравнений не имеют аналитических решений. В первую очередь это относится к большинству трансцендентных уравнений. Доказано также, что нельзя построить формулу, по которой можно было бы решить произвольное алгебраическое уравнение степени выше четвертой. Кроме того, в некоторых случаях уравнение содержит коэффициенты, известные лишь приблизительно, и, следовательно, сама задача о точном определении корней уравнения теряет смысл. Для их решения используются итерационные методы с заданной степенью точности. Итерационный процесс состоит в последовательном уточнении начального приближения х0. Каждый такой шаг называется итерацией. В результате итераций находится последовательность приближенных значений корня х1, х2, ..., хn. Если эти значения с увеличением числа итераций n приближаются к истинному значению корня, то говорят, что итерационный процесс сходится.

**Постановка задачи.** 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 (рис. 1).

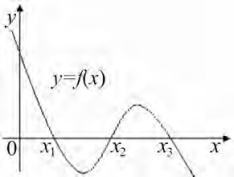


Рис. 1. График функции f(x)

Построение графиков часто удается сильно упростить, заменив уравнение (1) равносильным ему уравнением:

𝑓𝑓(𝑥1) = 𝑓𝑓(𝑥2) (2)

где функции 𝑓𝑓(𝑥1) и 𝑓𝑓(𝑥2) — более простые, чем функция f(x). Тогда, построив графики функций у = 𝑓𝑓(𝑥1) и у = 𝑓𝑓(𝑥2), искомые корни получим как абсциссы точек пересечения этих графиков (рис. 2). Точность решения невысока и определяется масштабом системы координат Оxy, в которой строятся графики функций (𝑥1) и 𝑓𝑓(𝑥2)

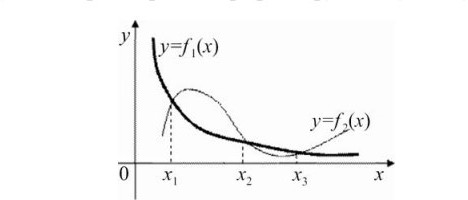


Рис. 2. Нахождение абсцисс точек пересечения графиков функций

Метод простой итерации.

Пусть уравнение (1) приведено к эквивалентному уравнению x = ϕ(x). (3)

Пусть известно приближенное, возможно весьма грубое, решение x0 исходного уравнения. Последующие приближения вычисляем по формуле

𝑥𝑛+1 = 𝜑𝜑(𝑥𝑛), n = 0, 1, 2,... (4)

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 (3) 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.

Метод итераций является одним из самых надежных методов решения уравнений.

Уравнение (1) можно преобразовать к виду (3) различными способами, например, полагая ϕ(x) = x + λf(x), где λ — некоторое число, выбираемое таким образом, чтобы выполнялись условия теоремы 1.

Пусть, например, на отрезке [a, b], где уравнение (1) имеет единственный корень, функция f(x) монотонна. Предположим для определенности, что f ′(x) > 0 на [a, b] (если f ′(x) < 0 на [a, b], то вместо уравнения (1) берем уравнение –f(x) = 0), тогда 0 < m < f ′(x) < M, где M и m — наибольшее и наименьшее значение функции f ′(x) на [a, b]. В этом случае полагаем

λ = 1

𝑀

*,* 𝛼 = 1 − 𝑚𝑚*.*

𝑀

|  |  |  |
| --- | --- | --- |
| **Лекционное занятие №7** | Итерационный метод Ньютона для решения нелинейного уравнения, геометрическая трактовка, оценка скорости сходимости. Итерационный метод Ньютона для систем нелинейных уравнений | **1** |

ПРИБЛИЖЕННОЕ РЕШЕНИЕ УРАВНЕНИЯ *F*(*х*) = 0

**МЕТОДОМ КАСАТЕЛЬНЫХ (Ньютона)**

Если известно начальное приближение решения уравнения *F*(*х*) = 0 на отрезке [*а, b*], то уточнить корень можно, как уже говорилось, любым способом. Одним из самых эффективных и точных решений является метод Ньютона (метод касательных), который состоит в построении итерационной последовательности

*xn*1  *xn*  *F* (*xn* ) *F*  (*xn* ) ,

сходящейся к корню уравнения *F*() = 0. Для применения этого метода необходимо существование первой и второй производных и их знакопостоянство на исследуемом отрезке.

Рассмотрим метод более подробно. Пусть найдено некоторое *xn =* , где   [*а, b*]. При

уточне25ое корня по методу Ньютона полагаем  *= xn* + *hn* , где *hn* - достаточно малое число. Тогда,

считая  корнем уравнения, находим *hn* , применив форму25о Тейлора:

0  *F* ()  *F* (*xn*  *hn* ) = *F* (*xn* ) + *hnF* (*xn* ) *,*

откуда *hn*   *F* (*xn* ) *F* ( *xn* ) , и окончательно получаем рекуррентную формулу

Достаточные условия сходимости определяются теоремой.

*xn*  *xn*1 

*f* (*xn*1 ) . (1)

*f* (*xn*1 )

**Теорема:** *Пусть F*(*х*) *определена и дважды дифференцируема на отрезке* [*а, b*]*, причем*

*F*(*а*)*F*(*b*) < 0*, а производные F'*(*х*) *и F»*(*х*) *сохраняют знак на отрезке* [*а, b*]*. Тогда, исходя из на-*

*чального приближения х0*  [*а, b*]*, удовлетворяющего неравенству F'*(*х0*)*F»*(*х0*) > 0, *можно по- строить последовательность*

*xn*1  *xn*  *F* (*xn* )

*F* (*xn* ) *, n* = 0, 1, *…,*

*сходящуюся к единственному на отрезке* [*а, b*] *решению*  *уравнения F*() *=* 0.

Если нулевое приближение выбрано достаточно близко к корню, то итерации сходятся очень быстро, со скоростью геометрической прогрессии.

Алгоритм данного метода очень простой и ясный и имеет, как и метод хорд, графическую интерпретацию (рис. 1).

1. Через точку *А*0(*х, у*) с координатами *х = b, у = F*(*b*) проводим касательную.
2. Находим пересечение касательной с осью 0*x* (точка *х*1).
3. Находим значение функции в точке *х = х*1; *y = F*(*х*1).
4. Проводим касательную в точке *х = х*1; *y = F*(*х*1) (точка *А*1).
5. Находим точку пересечения касательной с осью 0*х* (точку *х*2), и так далее до выполнения ус- ловия

Для оценки

*xn*  *xn*1   . За корень  принимаем *хn*. погрешности *n*-го

приближения корня [Березин, Жидков ] можно воспользоваться неравенством:

  *xn*  *xn*  *xn*1 *M*2 / (2*m*2 ) ,

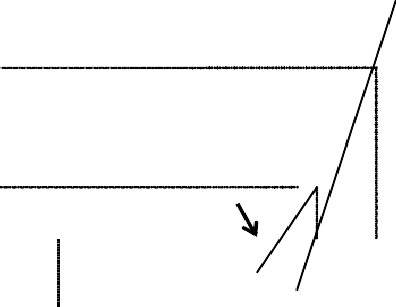
где *М2 =* sup *|F*»(*х*)*|* на отрезке [*а, b*], *m2 =* inf *|F'*(*х*)*|*

отрезке [*а, b*]. Таким образом, если

*xn*  *xn*1   , то   *xn*  *M*2  2 / (2*m*2 ) .

Последнее неравенство означает, что при удачном чальном приближении корня после каждой ите- рации количество верных цифр удваивается. Сле- вательно, процесс вычисления корня можно пре- кратить, если

Рис. 1. Геомет рическая инт ерпрет ация мет ода касат ельных (Ньют она)



у



F(b)



A0

0

* F(x1)

а





A1

   x 

b

x2

1

x

на

на-

до-

*xn*  *xn*1    .

2*m*2 */ M*2

Заметим еще раз, что метод Ньютона эффективен, если выбрано хорошее начальное при- ближение корня и график функции имеет большую крутизну в окрестности корня. В этом случае процесс быстро сходится. Если же численное значение *F'*(*х*) вблизи корня мало, т.е. график почти параллелен оси *0х,* то процесс вычисления корня будет долгим и ошиб26о округления чисел в машине могут вызвать обратный процесс – решение начнет расходиться. В этом случае можно по- советовать воспользоваться для уточнения корня другими более эффективными методами.

Метод Ньютона для систем нелинейных

Пусть дана система из n нелинейных уравнений с n неизвестными.

f1(x1,…,xn)=0,

f2(x1,…,xn)=0,

###### … … … fn(x1,…,xn)=0.

где fi(x1,…,xn) : Rn→R, i=1,…,n - нелинейные функции, определенные и непрерывно дифференцируемые в некоторой области G⊂Rn..

Запишем ее в векторном виде:

###### F(x)=0,

где x = (x1, x2, … , xn)Т, F(x)=[f1(x), f2(x), … , fn(x)]Т.

Требуется найти такой вектор x∗ = (x∗, x∗, … , x∗ )Т, который, при подстановке в исходную

1 2 𝑛

систему, превращает каждое уравнение в верное числовое равенство.

При таком подходе формула для нахождения решения является естественным обобщением формулы одномерного итеративного метода:

###### x(𝑘+1) = x(𝑘) − 𝑊𝑊−1�x(𝑘)�𝐹�x(𝑘)�, k=0,1,2,…,

Где

𝜕𝜕𝑓𝑓1(𝑥)

𝜕𝜕𝑥1

… 𝜕𝜕𝑓𝑓1(𝑥)

𝜕𝜕𝑥𝑛

𝑊𝑊 = �

… … …

� – матрица Якоби.

𝜕𝜕𝑓𝑓𝑛(𝑥)

𝜕𝜕𝑥1

… 𝜕𝜕𝑓𝑓𝑛(𝑥)

𝜕𝜕𝑥𝑛

|  |  |  |
| --- | --- | --- |
| **Лекционное занятие №8** | Постановка задачи интерполирования. Интерполяционный многочлен Лагранжа. Погрешность интерполирования, оценка погрешности. | **1** |

**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].

**Опеделение.** 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

 .

Интерполяционный многочлен в форме **Ньютона** имеет вид



где



. . . – выражения такого вида называются

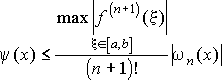
разделенными разностями .

**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 

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



где



Полиномы Эрмита

Полиномы Эрмита интерполируют таблично заданную функцию с учетом известных значений производной в узлах сетки.

Пусть заданы n+1 узлов xi , x0 = a, xn = b , x0 < x1 < x2 <….< xn , значения функции в них yi = f(xi) и значения производной в них yi ' = f' (xi) i = 0,…, n. Требуется построить полином P2n+1 (x) такой, что P2n+1(xi) = yi, P2n+1' (xi)= yi ' . Этот полином и называется полиномом **Эрмита**.

Интерполяция сплайнами

Пусть функция y=f(x) задана таблично :

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

Кубической сплайн-интерполяцией называется функция  (x) такая, что

 (xi) = f(xi) , i=0,1,…,n ,

 ’ (xi-0) =  ’ (xi+0) ,  ’ ’ (xi-0) =  ’ ’ (xi+0) i=1,...,n-1 (1)

 ’ ’ (x0) =0,  ’ ’ (xn) =0,

и  (x) = ai + bi(x-xi) +ci(x-xi)2 +di(x-xi)3 , xi-1  x  xi

Величины коэффициентов a,b,c,d, находятся из системы уравнений (1). Для нахождения значений этих коэффициентов удобно, с помощью последовательного исключения неизвестных, редуцировать систему (1) к системе трехточечных уравнений относительно коэффициентов сi , и решать ее далее с помощью метода прогонки.

|  |  |  |
| --- | --- | --- |
| **Лекционное занятие №9** | Интерполяционная формула Ньютона. Разделенные разности. Первая и вторая интерполяционные формулы Ньютона, оценки погрешности интерполирования. | **1** |

ПОСТРОЕНИЕ ИНТЕРПОЛЯЦИОННОГО ПОЛИНОМА НЬЮТОНА ПО ЗАДАННЫМ ЗНАЧЕНИЯМ ФУНКЦИИ

В силу единственности многочлена степени *n*, построенного по *n* + 1 значениям функции *f*(*х*), *многочлен Ньютона Рn*(*х*) является разновидностью записи интерполяционного многочлена и пол- ностью совпадает с многочленом, построенным по формуле Лагранжа. Однако с помощью много- члена Ньютона удобнее проводить интерполяцию в начале и конце таблицы значений функции. В начале таблицы используют *первую интерполяционную форму29о Ньютона*, а в конце –

*вторую*. Рассмотрим один из способов построения первой интерполяционной формулы.

Пусть некоторая функция *f*(*х*) задана табличными значениями *у*0 *= f*(*х*0)*; у*1*= f*(*х*1)*; …; уn = f*(*хn*) в равноотстоящих узлах интерполяции {*х*0 *, х*1 *= х*0 *+ h;…; хn = x*0*+ nh*}*.* Требуется построить интерполяционный полином Ньютона *Рn*(*х*) степени *n*, при котором

*Рn*(*х*0)  *у*0*; Рn*(*х*1)  *у*1*; …; Рn*(*хn*)  *уn .* (1)

Будем искать полином в виде

*Pn*(*x*0) *= a*0 *+ a*1(*x – x*0) *+a*2(*x – x*0) (*x – x*1) *+ ……+ an*(*x – x*0) (*x – x*1)*…*(*x – xn*)*,* (2)

где неизвестные коэффициенты *аi* находятся из очень простых зависимостей. Для того чтобы найти *а*0, положим *х=х*0. Очевидно, что при этом *Рn*(*х*0)  *у*0 *а*0 [это следует из формулы (12) ]. Но так как все члены уравнения (2) , кроме первого, содержат сомножитель (*x – x*0), следовательно, они все станут равными нулю, и из формулы (2) с учетом формулы (1) имеем *Рn*(*х*0) = *а*0  *у*0 .

Для того чтобы найти *а*1, положим *х = х*1*.* Повторив все рассуждения и учитывая, что значение полинома в указанной точке будет тождественно равно *у*1 [формула (2.2)], после подстановки в формулу (2.3) *х*1 имеем

*Pn*(*x*1) *= a*0 *+ a*1(*x*1 *– x*0) *= у*0 *+ а*1 *h = у*1 *.*

Все остальные сомножители при неизвестных коэффициентах *аi* будут равны нулю. Преобразуя последнее выражение, нахо30ое *а*1 как *а*1 *=* 1*/h* [здесь 1 *= Рn*(*х*0 *+ h*) *- -Рn*(*х*0) *= у*1

*– у*0 ].

Для того чтобы определить *а*2, положим *х = х2* и, рассуждая аналогично, определим третий коэффициент как *a*2 *=* 2 */* (2! *H2*)*.*

Подставляя в выражение (1) последовательно все *хn*, прихо30ое к общей формуле для

получения коэффициентов *аi*:

*ai =* *i /* ( *i*! *Hi*)*,*

которые подставим в формулу (2) и получим первую интерполяционную формулу Ньютона:

*Pn* ( *x*)  *y*0  1 *x*  *x*0   2 ( *x*  *x*0 )( *x*  *x*1)

1! *h*

…

2! *h*2

*n*1

*n*

*n* ( *x*  *xi* ) .

(3)

*n*! *h*

*i* 0

В формуле (3) обычно выполняют замену переменных: *q =* (*х – х*0)*/h,* где *h* – шаг интерпо- лирования. Тогда пер30ое30 интерполяционная формула Ньютона записывается несколько ина- че:

*Pn* ( *x*)  *y*0  1  *q*  2  *q*  (*q*  1) …

2!

(4)

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

*n*!

При *n* = 1 получаем формулу линейного интерполирования; при *n* = 2 – параболического интерполирования и т.д.

*Вторую интерполяционную формулу Ньютона* получают, если узлы интерполяции в *Рn*(*х*) берут в несколько ином порядке:

*Pn*(*x*0) = *a*0 + *a*1(*x – xn*) + *a*2(*x – xn*) (*x – xn*-1) + …

…+ *an*(*x – xn*) (*x – xn*-1) … (*x – x*0) .

Тогда, рассуждая как и в случае первой интерполяци30ое30 формулы, получаем искомую форму записи полинома Ньютона, которая известна как вторая интерполяционная формула:

*Pn* ( *x*)  *yn* 

1 ( *x*  *x* ) 

1! *h*

*n*

*n*

2

2! *h*2

*n*1

( *x*  *xn* )( *x*  *xn*1)

(5)

…

*n*! *h*

*n* ( *x*  *xi* ) .

*i* 0

Выполнив подстановку *q* = (*х – хn*)/*h*, получим иную запись 31ое31ия31м31ционного многочлена Ньютона:

*Pn* ( *x*)  *yn*  1  *q*  2  *q*  (*q*  1) …

2!

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

*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 интерполирова31ое *q* = (*х – хi*)/*h* > 0. При экстраполировании назад по первой интерполяци31ое31 формуле *q* = (*х – хi*)/*h* < 0. При интерполировании в конце таблицы, т.е. при интерполирова31ое назад, когда шаг интерполяции постоянен, применяют вто- рую интерполяционную форму31о Ньютона, где *q* = (*х – хi*+1) / *h* < 0. Эту же формулу используют при экстраполирова31ое вперед (в конце таблицы), но тогда *q* = (*х - -хi*+1)/ *h*>0.

|  |  |  |
| --- | --- | --- |
| **Лекционное занятие №10** | Квадратурная формула интерполяционного типа. Квадратурная формула Ньютона-Котеса. Оценки погрешности.  Квадратурные формулы прямоугольника, трапеции, Симпсона. | **1** |

Квадратурная формула интерполяционного типа.

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

*b*

   *f* *x**dx*

*a*

на указанном отрезке.

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

*n x*  *xi* ,

*Ln*   *yi*   *x*  *x*

*i*  0 *i*  *j j i*

который для равноотстоящих узлов примет вид

*n* 1*n**i*

*i*  0

*q**q*  1…*q*  *n* ,

где *q* = (*х – х*0) / *h*.

*Ln*   *yi*  *i*! *n*  *i*! 

*q*  *i*

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

*b b*

   *f* *x**dx*   *Lndx* 

*a a*

*b n* 1*n**i*

*q**q*  1…*q*  *n*

   *yi*  *i*! *n*  *i*! 

*a i*  0

*dx*.

*q*  *i*

Проведя необходимые элементарные преобразования, выполнив замену переменных *dq =*

*dx/n* и сменив в соответствии с подстановкой пределы интегрирования, получим

*b n* 1*n**i*

   *f* *x**dx*   *yi*  *i*! *n*  *i*!  *h* 

*a i*  0

*n*

  *q**q*  1…*q*  *i*  1  *q*  *i*  1…*q*  *n**dq* .

0

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

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

*b n*

   *f* *x**dx* *b*  *a*   *yi*  *Hi* ,

где

1*n**i* 1 *n*

*a i*  0

;

*Hi*  *i* !

*n*     *q**q*  1…*q*  *i*  1*q*  *i*  1…*q*  *n**dq*

 *i*! *n*

0

*Нi* – *коэффициенты Ньютона – Коте32о*. 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;

Квадратурные формулы прямоугольника и трапеции.

Пусть теперь некоторый конечный интервал [*а, b*] на оси *Ох* разбит на *n* подынтервалов [*хi, хi+1*], которые в дальнейшем будем называть элементарными отрезками. Ясно, что без ограничения общности можно положить *х*0 *= а; хn= b* и *х*0 *< <х*1 *< … <хn*. Через *hi* обозначим длину элементарного отрез32о (*хi*+1 *– хi*). Если заданный отрезок [*а, b*] разбит равномерно, то тогда *hi* будет постоянна для любой *i*  [*а, b*]. Пусть теперь на [*а, b*] определена некоторая функция *f*(*х*)*.* Предположим, что не-

*b*

обходимо найти приближение к определенному интегра32о, которое обозначим

 *f*    *f* *x**dx* .

*a*

Очевидно также, что если *f*(*х*) непрерывна *xi*  [*а, b*], то тогда

 *f* 

можно представить как

 *N* , где 

- интеграл функции *f*(*х*) на элементарном отрезке [*хi*+1*, хi*]*,* т.е.:

 *f*   *i i*

*i*  0

     

*i*1

 

*x*

.

*i i f*

 *f x dx*

*xi*

Bсякая простая формула, аппроксимирующая отдельный интеграл *i* , назывaется *квадтур-*

*ной*. *Составная квадратурная формула* – это формула, дающая приближение интеграла  *f*  в виде суммы приближений интегралами *i* по данной квадратурной формуле.

Двумя простейшими квадратурными формулами являются формулы *прямоугольников и*

*трапеций*, которые в ряде случаев оказываются наиболее эффективными.

Известны три разновидности формул прямоугольников: это формулы *левых, правых и средних прямоугольников*. Все они основаны на аппроксимации каждого интеграла *i* площадью прямоуголь32ое32, одной из сторон которого является *hi* , а второй – либо значение функции на

левом конце отрез33о (рис.3.1, *а*), либо значение функции на правом конце отрезка (рис. 3.1, *б*), либо значение функции в средней точке отрезка (рис. 3.1, *в*).

Квадратурные формулы, аппроксимирующие *i* , будут иметь вид: левых прямоугольников: *i = hi f* (*хi*)*;*

правых прямоугольников: *i = hi f* (*хi+*1)*;* средних прямоугольников: *i* = *hi f* (*хi*+1/2).

С учетом представления *i* на элементарном отрезке составные квадpатурные формулы прямоу- гольников могут быть записаны так:

левых прямоугольников

n-1 *n*1

*i*   *hi*  *f* *xi*  ;

правых прямоугольников

*i* =0

n

*i*0

*n*

*i*   *hi*  *f* *xi*1  ;

средних прямоугольников

*i* =1 *i*1

*n*-1 *n*1

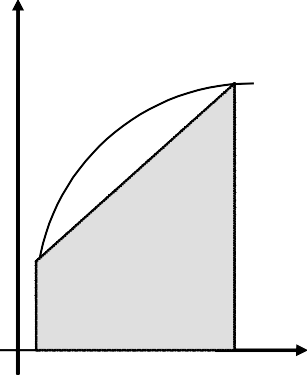
  *i*   *hi*  *f* *xi**h*/2  .

*i* =0

*i*0

Квадратурная формула трапеции.

В формуле трапеций используются значения функции в концевых точ- ках элементарных отрезков. В этом случае *i* аппроксимируется площадью трапеции с основаниями *f*(*хi*) и *f*(*хi*+1) и высотой *x* (рис. 3.2).



f(xi+1)

f(xi)

xi

xi+1

Тогда площадь фигуры мо33ое33 быть определена из формулы площа- ди прямоугольной трапеции

Рис. 1 Геомет рическая инт ер- прет ация “ мет ода т рапеций”

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

Если теперь просуммируем последнюю форму33о по всем элементар- ным отрезкам, то по33очим с учетом выполненных элементарных преобразо- ваний следующее выражение:

*n*

  *i*  *b*  *a*   *fa*  *fb*  / 2 

*i*1

*n*1

 *f* *xi*   *f* *xi*1  / *n* .

*i*2

Заметим, что при бесконечном уменьшении длин элементарных отрезков формулы обоих типов (прямоугольников и трапеций) сходятся к точному значению интеграла . Однако не ясно, как быстро они сходятся ? Попытаемся выяснить данный вопрос, воспользовавшись разложением функции в ряд Тейлора относительно центра элементарного отрезка [*хi, хi*+1]

*f*(*x*) *= f*(*yi*) *+* (*x – yi*) *f ’*(*yi*) *+* (*x – yi*)2 *f ’’*(*yi*)*/*2 *+*

*+* (*x – yi*)3 *f’’’*(*yi*)*/*6 *+* (*x-yi*)4 *f* (IV)(*yi*)*/*24 *+ …*

Затем, проинтегрировав полученный ряд по каждому из отрезков в предположении, что остав- шиеся члены ряда намного меньше выписанных, с учетом значений коэффициентов ряда Тейлора на элементарном отрезке

*Xi*1

*hi* , *p*  0 ;

0, *p*  1 ;



*x* - *y*  *p dx*  *h*3 / 12, *p*  2 ;

 *i*  *i*

*X* 0, *p*  3 ;

*i* 

 *i*

получим

*h*5 / 80, *p*  4,

*Xi*1

 *f* ( *x*)*dx*  *hi*  *f*  *yi*  

*Xi*

1 *h*3  *f*  *y*  

24

*i i*

 1 *h*5  *f*  *IV*   *y*   …

1920 *i i*

Член

1. *h*3  *f*  *y*  показывает ошибку формулы прямоугольников без учета членов более

24 *i i*

высокого поряд34о. Если теперь подставим в форму34о трапеций значения функции в точках *х* =

*хi* и *х* = *хi*+1, то получим

*Xi*1

 *f* ( *x*)*dx*  *hi Xi*

*f* *xi*   *f* *xi*1 

2

 1 *h*3  *f*  *y* …

12

*i i*

Можно видеть, что ошибки формул средних прямоугольников и трапеций одного поряд34о, т.е. дают почти одинаковую точность.

Квадратурная формула Симпсона и ее остаточный член.

Метод Симпсона часто называют в литературе *методом парабол*. Очевидно, что точность вычис- лений 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 значения *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:

*i* 

*Xi*1

 *f* *x**dx* 

*Xi*

*Xi*1

 *P*2 *x**dx* 

*Xi*

*Xi*1 

*y* 

 *y*  

   *yi* 



*Xi*

*i* *x*  *xi*  

*h*

1. *i*

2*h*2

*x*  *xi* *x*  *xi*1  *dx* 



 2*hyi* 

*y*  *h*

 4*h*2

2

  2  *yi* 

2*h*2

 8*h*3

3

  2  *yi*  

2*h*2

4*h*3 

2

 2*hyi*  2*h*( *yi* )  *h* 2  *yi*  / 3 

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

Так же, как и в § 2, просуммируем полученное

выражение по всем элементарным отрезкам, и если подставим *h* = =(*b – а*) / *n*, то окончательно получим

*N*

*i*   *h*  *yi*  4 *yi* 1/ 2  *yi* 1 / 6 

*i*  0

*b*  *a* 

*N N* 1 

 6*N*  *y*0  *yn*   4  *yi* 1/ 2  2  *yi* .

 *i*  0 *i* 1 

Данное выражение называется *формулой Симпсона*. Онo относится к формулам повыше- ной точности и является точной для многочленов второй и третьей степени [Бахвалов, 1973]. По- грешность формулы Симсона оценивается по формуле Тейлора и имеет вид [Данилина и др., 1976]

|  |  |  |
| --- | --- | --- |
| **Лекционное занятие №11** | Разностная аппроксимация дифференциальных операторов. Погрешность аппроксимации на сетке | **1** |

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

f ( xi )  fi , f '( xi )  f ', f ''( xi ) 

i

f ''.

Пусть функция задана в двух точках x0 и x1 

i

x0  h, ее значения f0 ,

f1.

35ое35ия35м интерполяционный многочлен первой степени

l1(x)  f0  (x  x0 ) f (x0;x1).

#### '

Производная l1(x) равна

l' ( x)  f ( x0; x1) 

1

f1 

# h

f0 .

Производную функцию f (x) в точке x0 приближенно заменяем производной интерполяционного многочлена

f ' (x) 

0

f1 

# h

f0 .

(1)

Величина

f1  f0

# h

называется первой разностной производной.

Пусть f (x) задана в трех точках x0 ,

x1 

x0  h,

x1 

x0  h.

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 x0 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

' f1 

f1

f0 

f1 

.

2h

f1

(2)

Величина

2h

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  

( 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 ,

получаем приближенную формулу.

'' f1  2 f0 

f



0 h2

f1 .

(3)

Величина

f1  2 f0 

h2

f1

называется второй разностной производной.

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

Предполагая функцию f достаточное число раз непрерывно дифференцируемой,

получим погрешности приближенных формул (1)-(3).

В дальнейшем нам понадобится следующая лемма.

**Лемма 1.** Пусть f Ca , b, i a , b 

произвольные точки, i  1, n.

Тогда существует такая точка  a, b, что

f (1) 

f (2 )… f (n ) 

# n

f ( ).

**Доказательство.** Очевидно неравенство

### min

a ,b

f ( x) 

f (1) 

f (2 )… f (n ) 

# n

max

a ,b

f ( x).

По теореме Больцано-Коши о промежуточных значениях непрерывной функции на

замкнутом отрезке она принимает все значения между min

a ,b

f ( x) и max

a ,b

f ( x). Значит

существует такая точка

 a, b, что выполняет указанное в лемме равенство.

Погрешности формул численного дифференцирования дает следующая лемма.

Лемма 2.

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

' f1 

f



0 h

f0  h

2

f ''

( ),

x0   

x1.

(4)

* 1. Если

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

# f ' 

f1 

f1  h2

f ''' (

x    x

0 2h 6

), 1 1.

(5)

* 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.

(6)

**Доказательство.** По формуле Тейлора

f  f  hf   h2 f   x x 

, , ,

#### 1 0 0 2 0 1

откуда следует (4).

Если

f C4x1, x1, то по формуле Тейлора

    *h*2

  *h*3

 *h*4

4  

(7)

*f* 1

*f*0 *hf*0

2 *f*0

6 *f*0 24 *f*  ,

где x1  

 

 x1.

Подставим (7) в

f1  2 f0 

h2

f1 . Получаем

f1  2 f0 

f1 

f 

h2  f  4 

  f  4 

.

h2 0 24  

Заменяя в соответствии с леммою 1

f 4   f 4   2 f 4, x1    x1,

получаем

*f*1

 2 *f*0

 *f*1 

  *h*2

*f* 4 .

*h*2 0 12

*f*

Откуда и следует (6).

Равенство (5) доказывается аналогично ( доказательство провести самостоятельно).

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 h (или порядка h ), а погрешность формул (2) и (3) имеет второй порядок относительно h (или

#### 2

порядка h ). Также говорят, что формула численного дифференцирования (1) первого порядка

точности (относительно h ), 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.

Выбор оптимального шага. Допустим, что граница абсолютной погрешности при вычислении функции f в каждой точке удовлетворяет неравенству

fi  . (8)

Пусть в некоторой окрестности точки x0 производные, через которые выражаются остаточные члены в формулах (5), (6), непрерывны и удовлетворяют неравенствам

4

f x  M3, f x  M4, (9)

где M3, M4 - некоторые числа. Тогда полная погрешность формул (2), (3) (без учета

погрешностей округления) в соответствии с (5), (6), (8), (9)не превосходит соответственно

величин

1 

    h2

2h 6

M3, (10)

2 

##   2  

h2

* h2

## 12

M4. (11)

Минимизация по h этих величин приводит к следующим значениям h :

## h   3 

1

3

1

4

, h  2 3 

, (12)

#### 1

при этом

 M3  2

#### 2

1

3



 M4 

1

2

1 

3  M3

##  



, 2

##  2

M4 

## 

. (13)

## 2  3 



 3 

Если при выбранном для какой-либо из формул (2), (3) значении h отрезок x1, x1 не выходит за пределы окрестности точки x0, в которой выполняется соответствующее неравенство (9), то найденное h есть оптимальным и полная погрешность численного дифференцирования оценивается соответствующей величиной (13).

**Опр**. 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) }.

**Опр.** 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

или .

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

Lh yh =  ( xh,yh) ( например, ).

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

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

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

Величины pk и  k выбираются из соображений аппроксимации.

|  |  |  |
| --- | --- | --- |
| **Лекционное занятие №12** | Метод конечных разностей решения задачи Коши для обыкновенных дифференциальных уравнений. Построение разностной схемы Эйлера. Понятие аппроксимации, сходимости, устойчивости разностной схемы | **1** |

При обсуждении методов решения задачи Коши ограничимся рассмотрением дифференциально- го уравнения первого порядка *у' = f*(*х,у*), удовлетворяющего начальному условию *у*(*х*0) *= у*0 *.*

Численное решение задачи состоит в построении таблицы приближенных значений *у*1*, у*2*,*

*…, уn*, являющихся решениями уравнения *у = у*(*х*) в точках *х*1*, х*2*, …, хn*. Чаще всего точки *хi* рас- положены равномерно: *хi = х*0 *+ h.i*, где *i = =*1, 2, *…, n.* Точ40о *хi* называют *узлами сетки, h – шагом сет40о*. Ясно, что *h >* 0*.*

Для получения решения задачи Коши воспользуемся разложением функции в ряд Тейлора. Для этого продифференцируем исходное уравнение *у' = f*(*х,у*) по *х n* раз. Тогда с учетом правила дифференцирования сложной функции получим следующие соотношения:

*y'' = fx* (*х, у*) *+ fy* (*х, у*)  *у'*;

*у''' = fxx* (*х,у*)*+2fxy*(*х,у*)  *у' + fyy*(*х,у*)  (*у'*)2 *+ fy*(*х,у*)  *у''*; . . .

Полагая *х = х*0 и *у = у*0, получаем ряд *у'*(*х*0); *у»*(*х*0); *у»'*(*х*0); … ; *у*(*n*)(*х*0), который сходится к решению поставленной задачи, т.е.

*n*

*y**x*   *y*

*i*  0

*i*

*x*0  

*x*  *x*0 *i* . (1)

*i*!

Надо заметить, что если *|х – х*0*|* больше радиуса сходимости ряда *у'*(*х*0)*; у»*(*х*0)*; у»'*(*х*0)*; …; у*(*n*)(*х*0)*,* то погрешность *|* *- у|* не стремится к нулю при *n*   , т.е. ряд расходится. Тогда посту- пают следующим образом. Отрезок [*х*0*,х*0*+Х*], на котором ряд расходится, разбивают на меньшие отрез41о [*хi, хi*+1]*,* где *i =* 1, 2, *…, n*, и получают новые последовательные приближения *уj* к ре- шению *у*(*хj* ) при *j* = 1, 2, *…, m*, используя следующую схему:

1. *уi* считаем найденным (для первого шага им мо41ое41 быть *у*0);
2. вычисляем в точке *хi* производные *уi*(*k*);
3. полагаем

*n*

*yx*  *zi* *x*   *y*

*k*  0

 *k* 

*x*0 

*x*  *x*0 *k* ;

*k* !

1. считаем *уi*+1 *= zi*(*хi*+1);
2. повторяем с первого пункта.

Если же в формуле (1) положить *n=*1, то получим основную расчетную формулу метода Эйле41о: *уi*+1*=уi+h* *f*(*х,у*)*.*

Этот метод относится к группе одношаговых, в которых для расчета точки (*хi*+1*, уi*+1) тре- буется информация только о последней вычисленной точке (*хi, уi*). Он допускает простую геометрическую интерпретацию. Получается, что на каждом новом приближении решение пере- ходит на другую кривую из семейства решений. Этот факт для некоторых дифференциальных уравнений может привести к большим ошибкам и решения задачи Коши начнет расходиться. Заметим, что хотя метод Эйле41о относится к итерационным, но ошибки, сделанные на ранних этапах, не уменьшаются из-за неустойчивости вычислительной схемы и ее чувствительности к шагу разбиения отрезка, на котором ищется решение. Чтобы обойти эту трудность, применяют другие вычислительные схемы или методы.

Для оценки погрешности метода Эйлера на одном из шагов сетки разложим точное решение в ряд Тейлора в окрестности узла *хi*:

*у(хi*+1*) = у(хi+h) = у(хi) + у'(хi).h +* *(h2) =*

*= у(хi) + f(хi,уi).h +* *(h2).*

Сравнение разложения с основной расчетной формулой метода показывает, что они совпадают до членов первого порядка по *h*, а погрешность формулы равна (*h*2), т.е. метод Эйлера относится к методам первого порядка.

Метод Эйлера относится к методам первого поряд41о, потому что обладает не только неустойчивостью решения, но и низкой точностью. Однако в смысле погрешности он дает удовлетворительные результаты даже при не очень больших *h*. К сожалению, ошиб41о при вы- чис41ое41 накапливаются от шага к шагу, и к концу отрезка решение содержит значительные погрешности. Заметим также, что источником ошибок очень часто являются и исходные данные. Существуют, однако, методы, позволяющие использовать все достоинства метода Эйлера и од- новременно повышающие точность вычислений. В нашей работе рассматривается несколько мо-

дифицированных методов Эйлера (расчетные схемы), по любому из них можно выполнять свои вычисления.

Рассмотрим основную расчетную формулу метода Эйлера: *уi*+1 *= уi+ h.f*(*х,у*). Одна из мо- дификаций 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.

|  |  |  |
| --- | --- | --- |
| **Лекционное занятие №13** | Методы Рунге-Кутта для решения задачи Коши для ОДУ первого порядка. Аппроксимация, погрешность аппроксимации,  порядок аппроксимации. Исследование порядка аппроксимации метода Рунге-Кутта 2-го порядка для ОДУ первого порядка | **1** |

МЕТОДЫ РУНГЕ–КУТТА

1. **Общая формулировка методов**

Формулировку методов Рунге–Кутта, ради простоты, рассмотрим на примере решения одного уравнения первого порядка, так как обобщение методов на систему уравнений не представляет трудностей и будет показано позже.

Итак, рассмотрим задачу Коши для уравнения

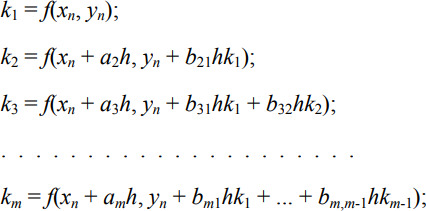
𝑑𝑢 = 𝑓𝑓(𝑥, 𝑢), 𝑥 > 𝑥

(1)

𝑑𝑥 0

𝑢(𝑥0)= 𝑢0

Явный m-этапный метод Рунге–Кутта заключается в следующем. Задаются некоторые числовые коэффициенты 𝑎𝑖𝑖 , 𝑏𝑖𝑖𝑖𝑖 , i = 2, 3, ..., т, j = 1, 2, ..., (m–1); 𝜎𝜎𝑖𝑖 , , i = 1, 2, ..., m и последовательно вычисляются следующие функции:



затем из формулы

(2)

находится решение задачи в следующей точке 𝑥𝑛+1: 𝑦𝑦( 𝑥𝑛+1) .

Коэффициенты 𝑎𝑖𝑖 , 𝑏𝑖𝑖𝑖𝑖 , 𝜎𝜎𝑖𝑖 выбираются из соображений точности аппроксимации разностным уравнением (2) дифференциального (1). Чтобы уравнение (2)

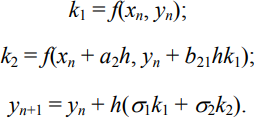
аппроксимировало (1), необходимо потребовать, чтобы ∑𝑚𝑚 𝜎𝜎𝑖𝑖 = 1 . Методы Рунге–

𝑖𝑖=1

Кутта при m > 5 используются нечасто. Наиболее широкое применение нашли методы Рунге–Кутта при m = 4.

При m = 1 получается одноэтапный метод — метод Эйлера . При m = 2 —

семейство двухэтапных методов следующего вида:



Рассмотрим погрешность аппроксимации двухэтапных методов в зависимости от выбора параметров: a, b, σ. Исключая из последнего уравнения функции k1 и k2 , сведем его к виду



(3)

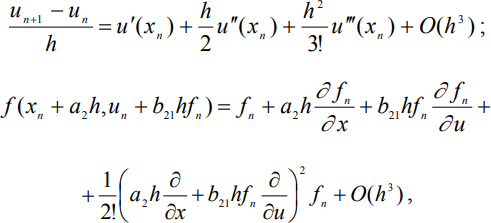
Невязка этого уравнения имеет вид



Здесь *u –* решение задачи Коши (1).

(4)

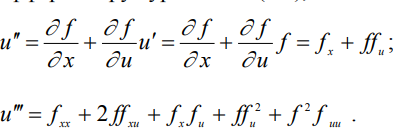
Предполагая достаточную гладкость решения u(х) и функции f(x, u), разложим все члены, входящие в выражение невязки, по формуле Тейлора в точке 𝑥𝑛 с точностью до O(h3 ):

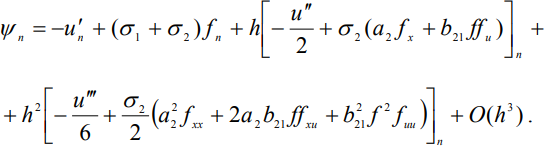


где



Последовательно дифференцируя уравнение (1), имеем:



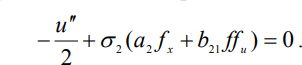
Подставляя полученные разложения функций и представления производных u″ и u″′ в выражение невязки (4) и собирая члены одинакового порядка, получаем

(5)

Из выражения (5) видно, что для того чтобы схема (3) имела первый порядок аппроксимации, необходимо выполнение условия



Тогда первые два члена невязки (5), с учетом уравнения (1), пропадают. Чтобы схема имела второй порядок аппроксимации, необходимо потребовать равенства нулю члена с h в первой степени:



Заменяя



это уравнение перепишем в виде



что выполняется при условиях



В результате получается однопараметрическое семейство двухэтапных методов Рунге– Кутта второго порядка аппроксимации при условии

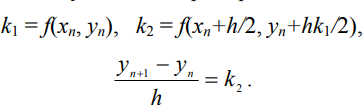


Это семейство можно записать в виде



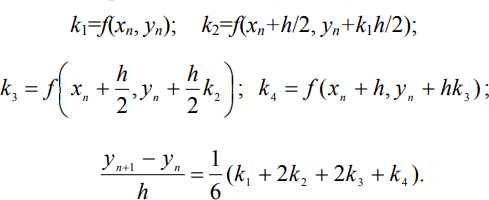
где σa = 0,5.

Например, при σ = 0,5, а = 1 получаем следующий метод Рунге-Кутта второго порядка



**Земечание.** Чтобы получить метод третьего порядка аппроксимации, необходимо приравнять к нулю сумму членов второго порядка малости в выражении для невязки (5)

На практике широко распространен **Метод Рунге – Кутта четвертого порядка :**



|  |  |  |
| --- | --- | --- |
| **Лекционное занятие №14** | Метод конечных разностей для решения краевой задачи для линейных ОДУ второго порядка. | **1** |

Рассмотрим линейное дифференциальное уравнение второго порядка

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

с краевыми условиями:

0  *y**a*  1  *y* *a*  *A;*

0  *y**b*  1  *y* *b*  *B.*



(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

y*xi*   *yi* 1  *yi* 1  *h*2  ;

2*h*

*y* *xi*   *yi* 1  2 *yi*  *yi* 1  *h*,

*h*2

а на концах отрезка – односторонними

*y* *x*0   *y*1  *y*0  *h* ;

*h*

*y* *xn*   *yn*  *yn*1  *h* .

*h*

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

 *yi*1  2 *yi*  *yi*1  *pi*  *yi*1  *yi*1  *qi*  *yi* 



*fi* ;

 2 2*h*

*h*



 *y*1  *y*0

(3)

 0 *y*0  1 

 *h*

 *A*;



0 *yn*  1 



*yn*  *yn*1  *B* .

*h*

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 по типу разло47ое47ия *конечно-разностным*. Однако матрица последней системы трехдиагональная, поэтому для ее ре- шения применима специальная 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 (5) относительно *уi* будет

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

Предположим, что *уi* уже найдено, тогда (6) 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

*y*1 

1 *y*1  *Ah*

  *n*0 *Ah*

 *f* 0*h*2  *y*2  .

*m*0  1  0*h*  *n*00     *h* 

 1 0



Сравнив последнее равенство с равенством (7), 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* :

*yi*  1   *fih*2  *nici*1*di*1  *yi*1  .

*m*  *n c* 

*i i i* 1

Сравним последнее выражение с равенcтвом (6), 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*



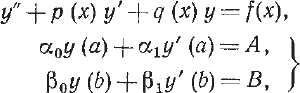
1

*n*1

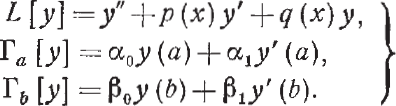
   0  *h*

|  |  |
| --- | --- |
| **Лекционное**  **занятие №15** | Метод Галеркина для решения краевой задачи для линейных ОДУ  второго порядка |

Рассмотрим краевую задачу для ОДУ 2-го порядка

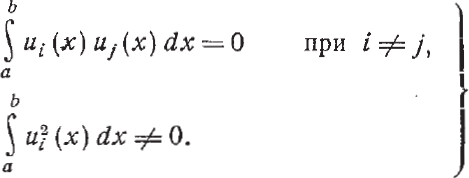


Обозначим



Пусть на [a,b] задана система базисных функций



Эта система по определению является ортогональной

Система базисных функций выбирается так, чтобы функция 𝑢0(𝑥) удовлетворяла неоднородным краевым условиям



а функции 𝑢𝑖𝑖(𝑥) удовлетворяли бы однородным краевым условиям

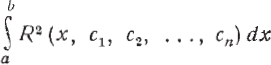


Решение краевой задачи ищем в виде



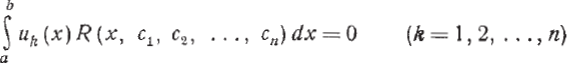
Рассмотрим выражение, называемое невязкой

Выберем коэффициенты с𝑖𝑖 таким образом, чтобы значение интеграла от невязки было наименьшим.

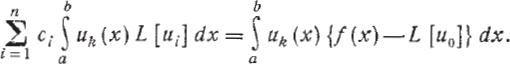


Это достигается лишь в том случае, если невязка ортогональна ко всем базисным функциям 𝑢𝑖𝑖.

Запишем условие ортогональности



или



Получили СЛАУ относительно коэффициентов с𝑖𝑖.