

Дисклеймер.

Автор не несет ответственности за любой ущерб, причиненный Вам при использовании данного документа. Автор напоминает, что данный документ может содержать ошибки и опечатки, недостоверную и/или непроверенную информацию. Если Вы желаете помочь в развитии проекта или сообщить об ошибке/опечатке/неточности:

[GitHub проекта](#)

[Автор в ВК](#)

# Содержание

1	Погрешность вычисления. Классификация погрешностей. Погрешности арифметических действий. Обратная задача теории погрешностей	4
2	Нормы матриц и векторов. Обусловленность матриц и систем линейных алгебраических уравнений. Примеры плохо обусловленных систем	5
3	Метод Гаусса решения СЛАУ. Проблемы практического применения. Использование для вычисления определителя, нахождения обратной матрицы	6
4	LU-разложение. Практическое применение, управление ростом погрешности — выбор ведущего элемента	8
5	QR-разложение, его сравнение с LU-разложением	9
6	Итерационные методы решения СЛАУ. Метод простой итерации. Условия сходимости, оценка погрешности	10
7	Итерационные методы решения СЛАУ. Метод Зейделя. Условия сходимости, оценка погрешности	12
8	Решение скалярных нелинейных уравнений. Метод половинного деления, метод хорд, метод секущих	13
9	Метод Ньютона решения нелинейных алгебраических уравнений и систем уравнений. Сходимость метода Ньютона. Модифицированный метод Ньютона	14
10	Алгебраическое интерполирование. Интерполяционный полином Лагранжа	16
11	Кратное интерполирование. Интерполяционный полином Эрмита	17
12	Интерполяция сплайнами. Общие сведения. Сплайн $S_{3,2}$	19
13	Интерполяция сплайнами. Общие сведения. Локальные (Эрмитовы) сплайны	20
14	Задачи численного интегрирования. Постановка задачи. Квадратурные формулы средних прямоугольников, трапеций и Симпсона. Оценка погрешности	22
15	Интерполяционные квадратурные формулы и квадратурные формулы Ньютона-Котса. Алгоритм построения. Оценка погрешности	22
16	Квадратурные формулы наивысшей алгебраической степени точности (типа Гаусса). Алгоритм построения. Оценка погрешности	22

17 Составные квадратурные формулы. Сходимость квадратурного процесса по составным квадратурным формулам. Составная квадратурная формула средних прямоугольников	22
18 Составные квадратурные формулы на основе формул с единичным весом (средних прямоугольников, трапеций, Симпсона). Оценка погрешности и алгебраическая степень точности	22
19 Практические способы оценки погрешности численного интегрирования составными квадратурными формулами (правило Рунге). Процесс Эйткена для оценки порядка аппроксимации	22
20 Методы типа Рунге-Кутты решения начальной задачи для ОДУ. Порядок и число этапов. Оценка методической погрешности на шаге	22
21 Практические способы оценки погрешности методов типа Рунге-Кутты. Вложенные методы	22
22 Алгоритм решения ОДУ методом типа Рунге-Кутты с автоматическим выбором шага	22

# 1 Погрешность вычисления. Классификация погрешностей. Погрешности арифметических действий. Обратная задача теории погрешностей

**Определение 1.1.** Погрешность — отклонение значения величины от её истинного (действительного) значения. Погрешность измерения является характеристикой точности измерения.

Погрешности бывают трех типов:

1) **Погрешность задачи**, связанная с приближенным характером исходной модели, ее математического описания, погрешностью измерений. При вычислениях подобная погрешность считается **неустранимой**.

2) **Погрешность метода**, связанная со способом решения поставленной математической задачи и появляющаяся в результате подмены исходной математической модели другой моделью. При создании численных методов закладывается возможность отслеживания подобных погрешностей и доведение их до сколь угодно малого уровня. Данная погрешность называется **устранимой**.

3) **Погрешность округления**, связанная с необходимостью выполнять арифметические действия над числами, усеченными до количества разрядов, зависящего от применяемой вычислительной техники.

В сумме все три погрешности дают **полную погрешность** результата решения задачи.

Пусть  $A$  — точное значение некоторой величины,  $a$  — ее приближенное значение.

**Определение 1.2.** Величину  $|A - a| = \Delta a$  называют абсолютной погрешностью.

**Определение 1.3.** Относительной погрешностью величины  $a$  называют отношение

$$\delta a = \frac{|A - a|}{|a|} = \frac{\Delta a}{|a|}$$

Рассмотрим погрешности арифметических действий. Пусть  $x_1 \pm \Delta x_1$  и  $x_2 \pm \Delta x_2$  — неточные числа. Тогда:

1) Сумма:

Абсолютная:  $(x_1 + x_2) + \Delta(x_1 + x_2) = x_1 + \Delta x_1 + x_2 + \Delta x_2 \Rightarrow \Delta_+ = \Delta x_1 + \Delta x_2$ , откуда  $|\Delta_{\pm}| \leq |\Delta x_1| + |\Delta x_2|$ .

Относительная:  $\frac{\Delta(x_1+x_2)}{x_1+x_2} = \frac{\Delta x_1}{x_1+x_2} + \frac{\Delta x_2}{x_1+x_2} \leq \delta x_1 + \delta x_2$ .

2) Вычитание:

Абсолютная аналогична сумме.

Относительная:  $\frac{\Delta(x_1-x_2)}{|x_1-x_2|} = \frac{\Delta x_1 + \Delta x_2}{|x_1-x_2|}$ , что указывает на возможность сильного возрастания погрешности при  $x_1 - x_2 \rightarrow 0$ .

3) Произведение:

Абсолютная:  $(x_1 + \Delta x_1)(x_2 + \Delta x_2) = x_1 x_2 + x_1 \Delta x_2 + x_2 \Delta x_1 + \Delta x_1 \Delta x_2 \Rightarrow \Delta_+ \approx x_1 \Delta x_2 + x_2 \Delta x_1$ .

Относительная:  $\frac{\Delta(x_1 x_2)}{x_1 x_2} \approx \frac{\Delta x_2}{x_2} + \frac{\Delta x_1}{x_1} \Rightarrow |\delta| \leq |\delta x_1| + |\delta x_2|$  (вытекает из вывода относительной).

4) Частное:

//

Абсолютная: ???

Относительная: ???

**Обратную задачу** теории погрешностей формулируют так: *каковы должны быть абсолютные погрешности аргументов функции, чтобы абсолютная погрешность функции не превышала заданной величины?*

**Пример 1.1.** 1) Принцип равных влияний: считаем, что вклад всех слагаемых в погрешность одинаков:

$$\left| \frac{\partial f}{\partial x_1} \right| \cdot \Delta x_1 = \left| \frac{\partial f}{\partial x_2} \right| \cdot \Delta x_2 = \dots = \text{const}$$

Откуда

$$\Delta x_i \leq \frac{|\Delta f|}{n \left| \frac{\partial f}{\partial x_i} \right|}$$

2) Принцип равных погрешностей: требуем одинаковых  $\Delta x_i$ :

$$\Delta x_1 = \Delta x_2 = \dots = \text{const} = \Delta x$$

Откуда

$$|\Delta x| \leq \frac{|\Delta f|}{\sum_{i=1}^n \left| \frac{\partial f}{\partial x_i} \right|}$$

## 2 Нормы матриц и векторов. Обусловленность матриц и систем линейных алгебраических уравнений. Примеры плохо обусловленных систем

**Определение 2.1.** Норма — функция  $\|\cdot\| : X \rightarrow \mathbb{R}$ , удовлетворяющая свойствам:

- 1)  $\|x\| \geq 0$ ,  $\|x\| = 0 \Leftrightarrow x = 0$ ;
- 2)  $\|\lambda x\| = |\lambda| \cdot \|x\|$ ;
- 3)  $\|x + y\| \leq \|x\| + \|y\|$ ;

**Пример 2.1.** Нормы векторов

$\|x\|_2 = \sqrt{x_1^2 + \dots + x_n^2}$  — «долгая» норма;

$\|x\|_1 = |x_1| + \dots + |x_n|$  — более простая норма;

$\|x\|_p = (|x_1|^p + \dots + |x_n|^p)^{1/p}$  — строгая математическая норма;

$\|x\|_\infty = \max_{i=\overline{1,n}} |x_i|$  — наиболее частоиспользуемая норма.

Все эти нормы эквивалентны.

**Определение 2.2.**  $\|\cdot\|_\alpha, \|\cdot\|_\beta$  эквивалентны, если  $\exists c_1, c_2: \forall x$  выполняется  $c_1 \|x\|_\beta \leq \|x\|_\alpha \leq c_2 \|x\|_\beta$ .

**Определение 2.3.** Рассмотрим линейный оператор  $A$ ; он ограничен, а значит,  $\|Ax\| \leq C$ . Тогда  $\min_x C = \|A\|$  — норма матрицы, согласованная с нормой вектора, если  $\|Ax\| \leq \|A\| \|x\|$ .

**Определение 2.4.** Норма матрицы, подчиненная норме вектора:

$$\|A\| = \sup_{\|x\| \neq 0} \frac{\|Ax\|}{\|x\|} = \sup_{\|x\|=1} \|Ax\|$$

**Пример 2.2.** Подчиненные нормы матриц:

- 1)  $\|A\|_1 = \max_{j=\overline{1,n}} \sum_{i=1}^n |a_{ij}|$ ;
- 2)  $\|A\|_2 = \sqrt{\max_{i=\overline{1,n}} \lambda(A^T A)}$ ;
- 3)  $\|A\|_\infty = \max_{i=\overline{1,n}} \sum_{j=1}^n |a_{ij}|$ .
- 4) Норма Фробениуса:  $\|A\|_F = \sqrt{\sum_{ij} a_{ij}^2}$ .

Пусть требуется решить СЛАУ  $Ax = b$ , где  $b \neq 0$ , причем правая часть системы содержит погрешности, то есть фактически требуется решить СЛАУ  $A(x + \Delta x) = b + \Delta b$ . Отсюда  $A(\Delta x) = \Delta b$ , и, считая матрицу  $A$  неособой, найдем  $\Delta x = A^{-1}\Delta b$ . Получим:

$$\|\Delta x\| = \|A^{-1}\Delta b\| \leq \|A^{-1}\| \|\Delta b\| = \|A^{-1}\| \cdot \|A\| \cdot \underbrace{\frac{\|b\|}{\|A\|}}_{=\|x\|} \cdot \frac{\|\Delta b\|}{\|b\|} \leq \nu(A) \|x\| \cdot \frac{\|\Delta b\|}{\|b\|}$$

где  $\nu(A) = \|A^{-1}\| \cdot \|A\|$ . Разделив обе части на  $\|x\| \neq 0$  получим  $\frac{\|\Delta x\|}{\|x\|} \leq \nu(A) \cdot \frac{\|\Delta b\|}{\|b\|}$ .

**Определение 2.5.** Величина  $\nu(A) = \|A^{-1}\| \cdot \|A\|$  называется числом обусловленности матрицы  $A$ .

**Определение 2.6.** Если  $\nu(A)$  намного больше единицы, СЛАУ называют плохо обусловленной.

Свойства числа обусловленности:

- 1)  $\nu(I) = 1$ ;
- 2)  $\nu(A) \geq 1$ :  $1 = \|E\| = \|A \cdot A^{-1}\| \leq \|A\| \|A^{-1}\| = \nu(A)$ ;
- 3)  $\nu(\alpha A) = \nu(A)$ ,  $\alpha \neq 0$ .

Пример плохо обусловленной системы:

$$\begin{pmatrix} 1 & 10 \\ 100 & 1001 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 11 \\ 11.01 \end{pmatrix}$$

### 3 Метод Гаусса решения СЛАУ. Проблемы практического применения. Использование для вычисления определителя, нахождения обратной матрицы

*Замечание 3.1.* Здесь и далее изучается вопрос о численном решении систем вида

$$\begin{cases} a_{11}x_1 + \dots + a_{1n}x_n = b_1 \\ \dots \\ a_{n1}x_1 + \dots + a_{nn}x_n = b_n \end{cases}$$

или иначе, векторно-матричных уравнений  $Ax = b$ .

Будем поэтапно приводить систему к треугольному виду, исключая последовательно сначала  $x_1$  из уравнений  $[2 : n]$ , затем  $x_2$  из  $[3 : n]$  и так далее.

На первом этапе заменим второе, третье и далее уравнения на уравнения, получающиеся сложением этих уравнений с первым, умноженным соответственно на  $-\frac{a_{21}}{a_{11}}, \dots, -\frac{a_{n1}}{a_{11}}$ . Результатом подобных преобразований будет система

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \\ a_{22}^{(1)}x_2 + \dots + a_{2n}^{(1)}x_n = b_2^{(1)} \\ \dots \\ a_{nn}^{(1)}x_n = b_n^{(1)} \end{cases}$$

коэффициенты которой имеют вид  $a_{ij}^{(1)} = a_{ij} - \frac{a_{i1}a_{1j}}{a_{11}}$ . При этом считаем, что  $a_{11} \neq 0$ , так как по условию матрица однозначно разрешима. Данные операции называются **прямым ходом** метода Гаусса.

Продолжая этот процесс далее на  $n - 1$ -ом этапе прямого хода приведем систему к треугольному виду

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \\ a_{22}^{(1)}x_2 + \dots + a_{2n}^{(1)}x_n = b_2^{(1)} \\ \dots \\ a_{nn}^{(n-1)}x_n = b_n^{(n-1)} \end{cases}$$

Нетрудно убедиться, что расчет коэффициентов производится по формулам:

$$a_{ij}^{(k)} = a_{ij}^{(k-1)} - \frac{a_{ik}^{(k-1)}}{a_{kk}^{(k-1)}}a_{kj}^{(k-1)}$$

$$b_i^{(k)} = b_i^{(k-1)} - \frac{a_{ik}^{(k-1)}}{a_{kk}^{(k-1)}}b_k^{(k-1)}$$

Треугольная структура матрицы позволяет последовательно вычислить значения неизвестных при помощи **обратного хода** метода Гаусса:

$$x_k = \frac{1}{a_{kk}^{(k-1)}} \left( b_k^{(k-1)} - \sum_{j=k+1}^n a_{kj}^{(k-1)}x_j \right)$$

Так как реальные машинные вычисления производятся не с точными, а с усеченными числами, то анализируя полученные формулы, можно сделать вывод о том, что выполнение алгоритма может привести к неверным данным или вовсе прекратиться, если знаменатели дробей на каком-то из этапов окажутся близкими или равными нулю. Чтобы избежать подобного рода ошибок выполняют перестановку строк таким образом, чтобы деление производилось на наибольший по модулю элемент столбца. Числа, на которые производится деление, называют **ведущими** или **главными элементами**. Отсюда название модификации: *метод Гаусса с постолбцовым выбором ведущего элемента*.

Устойчивость алгоритма к погрешностям исходных данных можно усилить, если выполнять выбор главного элемента по всей матрице. Такая модификация называется *методом главных элементов*, но применяется редко ввиду сложности алгоритма. Усложнение вызывает двумерный поиск и необходимость запоминать номера столбцов, к которым была применена перестановка, для обратной замены.

Так как определитель треугольной матрицы равен произведению элементов, стоящих на главной диагонали, то метод Гаусса может быть применен для поиска определителя матрицы. При этом, если алгоритм используется только для поиска определителя, из него следует убрать все шаги, в которых производится преобразование элементов векторов  $x$  и  $b$ .

## 4 LU-разложение. Практическое применение, управление ростом погрешности — выбор ведущего элемента

Пусть  $A$  — данная матрица размерности  $n \times n$ , а  $L$  и  $U$  — соответственно нижне- и верхнетреугольная матрицы размерности  $n \times n$ . Тогда справедлива

**Теорема 4.1.** *Если все главные миноры квадратной матрицы  $A$  отличны от нуля, то  $\exists L, U$  — нижне- и верхнетреугольная матрицы, такие, что  $A = LU$ . Если элементы диагонали одной из матриц  $L$  или  $U$  фиксированы, то такое разложение единственно.*

Рассмотрим

$$\begin{pmatrix} 1 & 0 & \cdots & 0 \\ l_{21} & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & \cdots & 1 \end{pmatrix} \begin{pmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & u_{nn} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}$$

Выполнив перемножение матриц, получим систему формул для вычисления элементов матрицы  $A$ . Специфика этой системы позволяет находить неизвестные  $l, u$  следующим образом:

$$u_{ij} = a_{ij} - \sum_{k=1}^{i-1} l_{ik} u_{kj}, \quad (i \leq j)$$

$$l_{ij} = \frac{1}{u_{jj}} \left( a_{ij} - \sum_{k=1}^{j-1} l_{ik} u_{kj} \right) \quad (i > j)$$

Препятствием для осуществления описанного процесса LU-разложения матрицы может оказаться равенство нулю диагональных элементов матрицы  $U$ , так как на них выполняется деление в последней формуле. Отсюда требование теоремы, накладываемое на главные миноры.

Для решения линейной системы уравнений мы можем переписать ее в виде  $LUx = b$  и, введя вектор вспомогательных переменных  $y$ , переписать в виде системы

$$\begin{cases} Ly = b \\ Ux = y \end{cases}$$

Таким образом, решение данной системы с квадратной матрицей коэффициентов свелось к последовательному решению двух систем с треугольными матрицами коэффициентов.



Получим формулы вычисления вектора  $y$ . Очевидно, что все они могут быть последовательно найдены по формуле

$$y_i = b_i - \sum_{k=1}^{i-1} l_{ik} y_k$$

Развернув теперь векторно-матричное уравнение  $Ux = y$  получим формулу для вычисления вектора  $x$ :

$$x_i = \frac{1}{u_{ii}} \left( y_i - \sum_{k=i+1}^n u_{ik} x_k \right)$$

Вычисление определителя LU-факторизованной матрицы  $A$  опирается на свойства определителя произведения матриц и определителя треугольной матрицы и сводится к нахождению определителя матрицы  $U$  уже известным методом.

Для обращения матрицы  $A$  при помощи LU-факторизации можно  $n$  раз использовать решение уравнения  $LUx = b_i$ , где  $b_i$  — столбцы матрицы  $I$ .

Если усложнить LU-разложение, добавив матрицы перестановок  $P$  и  $Q$ , такие, что осуществимо LU-разложение матрицы  $PAQ$  с выбором главного элемента по всей матрице, при этом  $P$  — матрица-перестановка строк, а  $Q$  — матрица-перестановка столбцов. Решение  $Ax = b$  происходит следующим образом: сначала полагаем  $x = Qz$ , тогда  $AQz = b$ . Умножив это равенство слева на матрицу  $P$ , приходим к эквивалентной системе  $PAQz = Pb$ , которая может быть представлено в виде  $LUz = Pb$ . Далее последовательно решаются треугольные системы  $Ly = Pb$ ,  $Uz = y$  относительно вспомогательных векторов  $y$  и  $z$  соответственно, после чего вычисляется искомый вектор  $x = Qz$ .

## 5 QR-разложение, его сравнение с LU-разложением

Выполняется декомпозиция  $A = QR$ , где  $R$  — верхнетреугольная матрица, а  $Q$  — ортонормированная.

**Определение 5.1.** Матрица поворота — матрица вида  $\begin{pmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{pmatrix}$ . Матрица обратного поворота, аналогично,  $\begin{pmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{pmatrix}$ .

**Метод вращений Гивенса:**

Повернем  $A_1$  на какой-то угол, чтобы первый столбец матрицы принял вид  $\begin{pmatrix} a_{11}^{(0)} \\ 0 \\ a_{31}^{(0)} \\ \vdots \\ a_{n1}^{(0)} \end{pmatrix}$ .

Для этого умножим матрицу  $A$  на матрицу  $Q_{21} = \begin{pmatrix} \cos \alpha & -\sin \alpha & 0 & 0 \\ \sin \alpha & \cos \alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$ . Это обнулит

элемент  $a_{21}$ . Аналогично, далее используем матрицу  $Q_{31} = \begin{pmatrix} \cos \alpha & 0 & -\sin \alpha & 0 \\ 0 & 1 & 0 & 0 \\ \sin \alpha & 0 & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$  для

обнуления элемента  $a_{31}$ . Произведение всех таких матриц будет являться матрицей  $Q = Q_{n,n-1}Q_{n,n-2}, \dots, Q_{n-1,n-1}, Q_{n-1,n-2}, \dots, Q_{21}$ .

Для нахождения угла  $\alpha$  используем следующее уравнение:

$$\begin{pmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{pmatrix} \cdot \begin{pmatrix} a_{11} \\ a_{21} \end{pmatrix} = \begin{pmatrix} a_{11}^{(1)} \\ 0 \end{pmatrix}$$

Легко вывести, что  $\begin{cases} \sin \alpha a_{11} + \cos \alpha a_{21} = 0 \\ \sin^2 \alpha + \cos^2 \alpha = 1 \end{cases}$ ,  $\sin \alpha = -\frac{\cos \alpha a_{21}}{a_{11}}$ ,  $\cos^2 \alpha + \cos^2 \alpha \left(\frac{a_{21}}{a_{11}}\right)^2 = 1$ ,

откуда  $\cos^2 \alpha = \frac{a_{11}^2}{a_{11}^2 + a_{21}^2}$ , отсюда  $\cos \alpha = \frac{a_{11}}{\sqrt{a_{11}^2 + a_{21}^2}}$ ,  $\sin \alpha = -\frac{a_{21}}{\sqrt{a_{11}^2 + a_{21}^2}}$ .

Плюсы в сравнении с LU-разложением: не нужно выбирать ведущий элемент и не растет вычислительная погрешность. Из минусов: работает в 4 раза медленнее.

//попросить его рассказать про данный метод подробнее

**Метод отражений Хаусхольдера:**

Рассмотрим вспомогательный вектор  $\omega$  — вектор единичной длины, такой, что  $\omega^T \omega = 1$ .

Рассмотрим  $U = E - 2\omega\omega^T$ ,  $U^T U = E - 4\omega\omega^T + 4\omega \underbrace{\omega^T \omega}_1 \omega^T = E \Rightarrow U^{-1} = U^T$ .

$U_\omega = (E - 2\omega\omega^T)\omega = \omega - 2\omega = -\omega \Rightarrow \omega$  — собственный вектор с собственным числом  $-1$ .

$v \perp \omega$ , то есть  $v^T \omega = 0$  или  $\omega^T v = 0$ ,  $U_v = (E - 2\omega\omega^T)v = v - 2\omega\omega^T v = v \Rightarrow v$  — собственный вектор с собственным числом  $1$ .

Таким образом,  $y = v + \alpha\omega \Rightarrow Uy = v - \alpha\omega$ , то есть матрица  $U$  отражает вектор.

Пусть  $y, z$  — ... векторы. Нам нужно найти  $U$ , такую, что  $Uy = \alpha z$ . Смотрим:

$$\|Uy\| = \|y\| = \|\alpha z\| \Rightarrow \alpha = \frac{\|y\|}{\|z\|}$$

$$\omega = \frac{y - \alpha z}{\|y - \alpha z\|}$$

Теперь, используя  $A_1$  как  $y$ ,  $e_1$  как  $z$ , строим  $U_1 = E - 2\omega\omega^T$ . Тогда  $U_1 A$  будет иметь нулевой первый столбец (исключая элемент  $a_{11}^{(1)}$ ).

Тогда  $Q = U_{n-1} \cdot \dots \cdot U_1$ .

Тогда решением уравнения  $Ax = b$  будет являться решение уравнения  $Rx = Q^T b$ , которое можно найти, применив формулу из LU-разложения (формула обратной подстановки).

## 6 Итерационные методы решения СЛАУ. Метод простой итерации. Условия сходимости, оценка погрешности

Рассматриваемая нами система  $Ax = b$  путем неких изменений может быть приведена к форме  $x = Bx + c$ . Тогда задача нахождения  $x$  становится задачей нахождения неподвижной точки отображения.

$$x^{(k+1)} = Bx^{(k)} + c$$

Интуитивно понятно, что  $\|B\| > 1$  влечет  $\|x_k\| \rightarrow \infty$  (так как в этом случае отображение не является сжимающим).

**Лемма 6.1.** Все собственные числа матрицы  $B$  по модулю меньше единицы тогда и только тогда, когда

- 1)  $B^k \rightarrow_{k \rightarrow \infty} 0$ ;
- 2)  $\exists (I - B)^{-1} = (I + B + B^2 + \dots + B^k + \dots)$ ;

**Лемма 6.2.** Если  $\|B\| \leq q < 1$ , то  $(I - B)^{-1}$  существует, равна  $\sum_{i=0}^{\infty} B^i$  и  $\|(I - B)^{-1}\| \leq \frac{1}{1-q}$ .

*Доказательство.*  $\|B\| \leq q < 1 \Rightarrow \|I + B + \dots + B^k + \dots\| \leq \|I\| + \|B\| + \dots + \|B^k\| + \dots \leq \|I\| + \|B\| + \dots + \|B\|^k + \dots \leq 1 + q + \dots + q^k + \dots = \frac{1}{1-q}$ , следовательно, существует  $V = \sum_{k=0}^{\infty} B^k$ , такое, что  $\|V\| \leq \frac{1}{1-q}$ .

Тогда  $(I - B)V = IV - BV = I + B + B^2 + \dots + B^k + \dots - B - B^2 - \dots - B^k - \dots = I$ , следовательно  $V = (I - B)^{-1}$  и  $\|(I - B)^{-1}\| \leq \frac{1}{1-q}$ .  $\square$

**Теорема 6.1.** Необходимым и достаточным условием сходимости метода простой итерации с любым начальным приближением  $x^{(0)}$  к  $x^*$ :  $x^* = Bx^* + c$  является ограниченность собственных чисел матрицы  $B$  числом, меньшим единицы.

*Доказательство.*

*Достаточность:*

$$x^{(1)} = Bx^{(0)} + c$$

$$x^{(2)} = B(Bx^{(0)} + c) + c = B^2x^{(0)} + Bc + c$$

...

$$x^{(k)} = B^kx^{(0)} + (B^{(k-1)} + \dots + I)c$$

Из условий: 1)  $B^k \rightarrow_{k \rightarrow \infty} 0$ ; 2)  $\exists (I - B)^{-1} = \sum_{i=0}^{\infty} B^i$  следует, что при  $k \rightarrow \infty$  выполняется  $x^{(k)} \rightarrow (I - B)^{-1}c$ .

Преобразуем  $x = Bx + c \Leftrightarrow (I - B)x = c$ , следовательно,  $(I - B)^{-1}c$  является решением.

Пусть  $\exists x^{**}$  — другое решение. Тогда  $x^* = Bx^* + c$ ,  $x^{**} = Bx^{**} + c$  и  $x^* - x^{**} = B(x^* - x^{**})$ , следовательно,  $\lambda = 1$  является собственным числом  $B$  с собственным вектором  $x^* - x^{**}$ , а это противоречит теореме.

*Необходимость:*

Так как  $x^{(k+1)} = Bx^{(k)} + c = B^{k+1}x^{(0)} + (I + B + B^2 + \dots + B^k)c$ , то из существования  $\lim_{k \rightarrow \infty} x^{(k+1)}$  вытекает существование предела правой части, что влечет сходимость матриц  $B^{k+1}$  к  $I$  и сходимость ряда  $\sum_{k=0}^{\infty} B^k$  к  $(I - B)^{-1}$ , что по лемме 6.1 влечет ограниченность собственных чисел матрицы числом меньше единицы.  $\square$

**Теорема 6.2.** Пусть  $\|B\| \leq q < 1$ , тогда МПИ (метод простой итерации) сходится  $\forall x^{(0)}$  к  $x^*$ :  $x^* = Bx^* + c$  и верны следующие оценки:

- 1)  $\|x^* - x^{(k)}\| \leq \frac{q}{1-q} \|x^{(k)} - x^{(k-1)}\|$  — апостериорная оценка;
- 2)  $\|x^* - x^{(k)}\| \leq \frac{q^k}{1-q} \|x^{(1)} - x^{(0)}\|$  — априорная оценка.

*Доказательство.*  $x^{(k-1)} - x^{(k)} = Bx^{(k)} + c - Bx^{(k-1)} + c = B(x^{(k)} - x^{(k-1)}) \Rightarrow \|x^{(k+1)} - x^{(k)}\| \leq q \|x^{(k)} - x^{(k-1)}\|$

$$\|x^{(k+m)} - x^{(k)}\| = \|x^{(k+m)} - x^{(k+m-1)} + \dots + x^{(k+1)} - x^{(k)}\| \leq \|x^{(k+m)} - x^{(k+m-1)}\| + \dots + \|x^{(k+1)} - x^{(k)}\| \leq (q^{m-1} + q^{m-2} + \dots + q + 1) \|x^{(k+1)} - x^{(k)}\| = \frac{1-q^m}{1-q} \|x^{(k+1)} - x^{(k)}\|$$

$$\text{Тогда } \|x^* - x^{(k)}\| \leq \frac{1}{1-q} \|x^{(k+1)} - x^{(k)}\| \leq \frac{q}{1-q} \|x^{(k)} - x^{(k-1)}\|.$$

Применив много раз, получим  $\|x^{(k+1)} - x^{(k)}\| \leq q^2 \|x^{(k-1)} - x^{(k-2)}\|, \dots, \|x^{(k+1)} - x^{(k)}\| \leq q^k \|x^{(1)} - x^{(0)}\|$ . Таким образом, при  $k \rightarrow \infty$  последовательность сходится в себе,  $\mathbb{R}^n$  полное, следовательно, существует  $x^* = \lim_{k \rightarrow \infty} x^{(k)}$ .

$(I - B)(B^k x^{(0)} + (I - B)^{-1}c) = c$ ,  $B^k x^{(0)} = 0$ , так как  $\|B\| \leq q \Rightarrow \|B^k\| \leq q^k$ .

И в результате  $\|x^* - x^{(k)}\| \leq \frac{q}{1-q} \|x^{(k)} - x^{(k-1)}\|$ . □

*Замечание 6.1.* Как правило, априорная оценка грубее апостериорной.

**Метод Якоби** приведения СЛАУ  $Ax = b$  к виду  $x = Bx + c$ :

Представим матрицу  $A$  в виде  $A = L + D + R$ , где  $L, R$  — соответственно строго (с нулевой диагональю) лево- и правотреугольные матрицы, а  $D$  — диагональная. Тогда мы можем переписать систему в виде  $Lx + Dx + Rx = b \Leftrightarrow x = -D^{-1}(L + R)x + D^{-1}b$  (должно выполняться условие на ненулевую диагональ матрицы  $D$ ). Таким образом, следует положить  $B = -D^{-1}(L + R)$ ,  $c = D^{-1}b$ .

Чтобы записать метод Якоби, достаточно заметить, что обратной к матрице  $D$  служит диагональная матрица с элементами  $1/a_{ii}$ . Таким образом, для получения матрицы  $B$  нужно каждую строку матрицы разделить на диагональный элемент в этой строке. Тогда в каждой строке получим  $x$  с соответствующим номером с единичным коэффициентом.

$$x_1 = -\frac{a_{12}}{a_{11}}x_2 - \dots - \frac{a_{1m}}{a_{11}}x_m + \frac{b_1}{a_{11}}$$

$$x_2 = -\frac{a_{21}}{a_{22}}x_1 - \frac{a_{23}}{a_{22}}x_3 \dots - \frac{a_{2m}}{a_{22}}x_m + \frac{b_2}{a_{22}}$$

и так далее. Соответственно, метод Якоби есть МПИ в такой системе.

**Определение 6.1.** Матрица с диагональным преобладанием — такая, что

$$\forall i = \overline{1, n} : |a_{ii}| > \sum_{j=1, j \neq i}^n |a_{ij}|$$

**Теорема 6.3.** Для матрицы с диагональным преобладанием метод Якоби сходится.

**Теорема 6.4.** Метод Якоби сходится тогда и только тогда, когда все корни уравнения

$$\det \begin{pmatrix} \lambda a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & \lambda a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & \lambda a_{nn} \end{pmatrix} = 0$$

по модулю меньше единицы.

//нужны ли доказательства?

## 7 Итерационные методы решения СЛАУ. Метод Зейделя. Условия сходимости, оценка погрешности

//нужно ли говорить здесь начало предыдущего билета?

Грубо говоря, метод Зейделя состоит в применении для вычисления  $i$ -ой компоненты на  $(k + 1)$ -ом шаге компонент, уже вычисленных на этом шаге ранее:

$$\begin{cases} x_1^{(k+1)} = b_{11}x_1^{(k)} + b_{12}x_2^{(k)} + \dots + b_{1n}x_n^{(k)} + c_1 \\ x_2^{(k+1)} = b_{21}x_1^{(k+1)} + b_{22}x_2^{(k)} + \dots + b_{2n}x_n^{(k)} + c_2 \\ \dots \\ x_n^{(k+1)} = b_{n1}x_1^{(k+1)} + b_{n2}x_2^{(k+1)} + \dots + b_{nn}x_n^{(k)} + c_n \end{cases}$$

Чаще всего используется метод, являющийся модификацией метода Якоби, расчеты в котором производятся по следующей формуле:

$$\begin{cases} x_1^{(k+1)} = - \left( a_{12}x_2^{(k)} + a_{13}x_3^{(k)} + \dots + a_{1n}x_n^{(k)} - b_1 \right) / a_{11} \\ x_2^{(k+1)} = - \left( a_{21}x_1^{(k+1)} + a_{23}x_3^{(k)} + \dots + a_{2n}x_n^{(k)} - b_2 \right) / a_{22} \\ \dots \\ x_n^{(k+1)} = - \left( a_{n1}x_1^{(k+1)} + a_{n2}x_2^{(k+1)} + \dots + a_{nn}x_{n-1}^{(k+1)} - b_n \right) / a_{nn} \end{cases}$$

где  $x^{(0)} = \left( x_1^{(0)}, \dots, x_n^{(0)} \right)^T$ .

**Теорема 7.1.** *Для сходимости метода Зейделя необходимо и достаточно, чтобы все корни уравнения*

$$\det \begin{pmatrix} \lambda a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & \lambda a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & \lambda a_{nn} \end{pmatrix} = 0$$

*были по модулю меньше единицы.*

*Доказательство.* Преобразуем ход метода Якоби к новым условиям:  $Lx^{(k+1)} + Dx^{(k+1)} + Rx^{(k)} = b$ , откуда  $x^{(k+1)} = -(L + D)^{-1}Rx^{(k)} + (L + D)^{-1}b$ , что есть ничто иное, как МПИ.

Применяя теорему 6.1 к этому МПИ, составим характеристическое уравнение, определяющее собственные числа матрицы  $B = -(L + D)^{-1}R$ :  $\det(-(L + D)^{-1}R - \lambda I) = 0 \Leftrightarrow \det((L + D)\lambda + R) = 0$ , которое, с учетом  $A = L + D + R$ , совпадает с условием теоремы.  $\square$

**Теорема 7.2.** *Пусть  $\|(L + D)^{-1}R\| \leq t < 1$ . Тогда при любом начальном векторе  $x^{(0)}$  метод Зейделя сходится к решению  $x^*$  системы и справедливы оценки погрешности:  $\|x^* - x^{(k)}\| \leq \frac{t}{1-t} \|x^{(k)} - x^{(k-1)}\| \leq \frac{t^k}{1-t} \|x^{(1)} - x^{(0)}\|$ .*

**Теорема 7.3.** *Если в матрице  $A$  имеет место диагональное преобладание, то метод Зейделя сходится, причем быстрее, чем метод Якоби.*

//Нужно ли доказательство?

## 8 Решение скалярных нелинейных уравнений. Метод половинного деления, метод хорд, метод секущих

### Метод половинного деления

Пусть требуется найти корень уравнения  $f(x) = 0$ , где  $f$  определена  $[a, b]$ , на котором есть корни, то есть  $f(a)f(b) < 0$ . Для этого вычислим пробную точку  $c = \frac{a+b}{2}$ , вычислим  $f(c)$  и сравним:

- 1)  $f(a) - f(c) < 0 \Rightarrow b := c, f(b) := f(c)$  и рекуррентно выполняем.
- 2)  $f(a) - f(c) > 0$ . Все то же самое, но наоборот.
- 3)  $f(a)f(c) = 0$  — вернуть  $c$ .

Делаем, пока  $\frac{b-a}{2} > \varepsilon$ .

### Метод хорд

Будет более оптимально при вычислении  $c$  учитывать величины ординат  $f(a), f(b)$ . Запишем уравнение прямой, проходящей через эти точки:

$$\frac{y - f(a)}{f(b) - f(a)} = \frac{x - a}{b - a}$$

Отсюда, полагая  $y = 0$ ,  $x = c$  находим:

$$c = a - \frac{f(a)(b - a)}{f(b) - f(a)}$$

Данный метод также выполняется итерационно до достижения определенной точности.

### Метод секущих

Введем величину  $h_k = x_{k-1} - x_k$ , откуда  $x_{k-1} = x_k + h_k$ . Так как (общая формула)

$$f'(x) \approx \frac{f(x_k + h_k) - f(x_k)}{h_k}$$

и общий итерационный процесс метода Ньютона имеет вид

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

то подставив первое во второе получим

$$x_{k+1} = x_k - \frac{f(x_k)h_k}{f(x_k + h_k) - f(x_k)}$$

откуда

$$x_{k+1} = x_k - \frac{f(x_k)(x_{k-1} - x_k)}{f(x_{k-1}) - f(x_k)}$$

что является основным итерационным процессом метода секущих.

## 9 Метод Ньютона решения нелинейных алгебраических уравнений и систем уравнений. Сходимость метода Ньютона. Модифицированный метод Ньютона

Пусть  $x_k$  — уже известный член последовательности приближений. Для корня  $x^*$  можно записать формальное представление по формуле Тейлора:

$$f(x^*) = f(x_k) + f'(x_k)(x^* - x_k) + f''(\Theta) \frac{(x^* - x_k)^2}{2}$$

где  $\Theta \in [a, b]$ . Считая, что значение  $x_k$  близко к  $x^*$ , можно рассчитывать, что величина  $(x^* - x_k)^2$  будет малой. На этом основании отбросим ее и подменим квадратное уравнение линейным. Естественно, что при этом будет найден не корень  $x^*$ , а некоторая точка  $x_{k+1}$ , близкая к корню. Тогда итерационный процесс метода Ньютона определяется уравнением

$$f(x_k) + f'(x_k)(x_{k+1} - x_k) = 0$$

откуда можно вывести явный вид:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

Обозначим  $x_k - x_{k+1} = S(x) = \frac{f(x)}{f'(x)}$ . Если  $|S'(x)| \leq q \leq 1$  в некоторой окрестности, то МПИИ сходится.

$$S'(x) = -\frac{(f'(x))^2 - f(x)f''(x)}{(f'(x))^2} + 1 = -1 + \frac{f(x)f''(x)}{(f'(x))^2} + 1$$

то есть  $S'(x) = \frac{f(x)f''(x)}{(f'(x))^2}$ . Таким образом, условие сходимости имеет вид:

$$\left| \frac{f(x)f''(x)}{(f'(x))^2} \right| < 1$$

**Теорема 9.1.** Если на отрезке  $[a, b]$  существует корень,  $f'(x) \neq 0$  на  $[a, b]$  и  $f''(x) \neq 0$ , то метод Ньютона сходится  $\forall x_0$ , такого, что  $f(x_0)f''(x_0) > 0$ .

Оценим скорость сходимости:

$$f(x_k) + f'(x_k)(x^* - x_k) = F(x_k)$$

$$F(x) = f(x) + f'(x)(x^* - x); \quad F(x^*) = 0$$

$$F(x_k) = F(x^*) + \int_{x^*}^{x_k} F'(x)dx$$

$$F'(x) = f'(x) + f''(x)(x^* - x) + f'(x)(-1) = f''(x)(x^* - x)$$

$$F(x_k) = \int_{x^*}^{x_k} f''(x)(x^* - x)dx = f''(\xi) \int_{x^*}^{x_k} (x^* - x)dx = f''(\xi) \frac{(x^* - x_k)^2}{2}$$

$$x^* - x_{k+1} = x^* - x_k + \frac{f(x_k)}{f'(x_k)}$$

$$f'(x)(x^* - x_{k+1}) = F(x_k)$$

$$|x^* - x_{k+1}| \leq \frac{M_2}{f'(x_k)} |x^* - x_k|^2$$

таким образом,

$$|x^* - x_{k+1}| \leq \frac{q^{2^{(k+1)}-1}}{1-q} |x^* - x_0|$$

Цель изменения основной формулы метода Ньютона — уменьшение вычислительных затрат. Самый простой выход — использование на каждом шаге фиксированного множителя  $\frac{1}{f'(x_0)}$ , то есть осуществлять счет по формуле

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_0)}$$

Такой метод называют **модифицированным методом Ньютона**. Однако он имеет существенный недостаток — утрату высокой скорости сходимости.

## 10 Алгебраическое интерполирование. Интерполяционный полином Лагранжа

Пусть  $f(x)$  задана на  $[a, b]$  и  $f(x)$  — «сложная» или таблично задана в точках  $\{\bar{x}_1, \dots, \bar{x}_n\}$ . Задача интерполирования — быстро вычислять приближение к  $f(x)$  или просто  $f(x)$  в  $x \notin \{\bar{x}_1, \dots, \bar{x}_n\}$ . Для этого  $f$  заменяется на  $a_0\varphi_0 + \dots + a_n\varphi_n$ , где  $\varphi_0, \dots, \varphi_n$  — «простые» функции, причем  $a_0\varphi_0(x_i) + \dots + a_n\varphi_n(x_i) = f(x_i)$  для некоторых  $\{x_0, \dots, x_n\} \subset [a, b]$  или из  $\{\bar{x}_1, \dots, \bar{x}_n\}$ . Такой прием называется интерполяцией.

Интерполяционный полином  $a_0\varphi_0 + \dots + a_n\varphi_n$  может быть построен для заданной точности, только если функции  $\varphi_0, \dots, \varphi_n$  обладают следующими свойствами:

1)  $\varphi_0, \dots, \varphi_n, \dots$  — линейно независимы, то есть любой конечный набор из них линейно независим.

2)  $\varphi_0, \dots, \varphi_n, \dots$  — полная система функций в некотором рассматриваемом функциональном пространстве, то есть  $\forall f \exists b_0, \dots, b_n, \dots: f = \sum_{i=0}^{\infty} b_i\varphi_i$ .

**Определение 10.1.** Задача интерполяции: для функции  $f(x)$ , заданной таблицей, найти многочлен  $P_n(x)$  такой, что выполняется совокупность условий  $P_n(x_i) = f(x_i)$ .

Данная задача решается посредством решения СЛАУ

$$\begin{pmatrix} \varphi_0(x_0) & \varphi_1(x_0) & \cdots & \varphi_n(x_0) \\ \varphi_0(x_1) & \varphi_1(x_1) & \cdots & \varphi_n(x_1) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_0(x_n) & \varphi_1(x_n) & \cdots & \varphi_n(x_n) \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} f(x_0) \\ f(x_1) \\ \vdots \\ f(x_n) \end{pmatrix}$$

Однако делать это напрямую неэффективно. Поэтому будем строить

### Интерполяционный полином Лагранжа

Будем обозначать такой полином степени  $n$  как  $L_n$ .

Итак, должно выполняться  $L_n(x_i) = f(x_i)$ . Представим  $L_n(x_i) = l_i(x)f(x_i)$ , где  $l_i(x_j) =$

$$\begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \text{ — символ Кронекера.}$$

Для того, чтобы построить такой полином, нам нужно, чтобы у  $l_i(x)$  корнями являлись  $x_0, \dots, x_{i-1}, x_{i+1}, \dots, x_n$  (чтобы значение  $l_i(x_0, \dots, x_{i-1}, x_{i+1}, \dots, x_n)$  были нулевыми. Тогда (знаменатель нужен для того, чтобы  $l_i(x_i) = 1$ )

$$l_i(x) = \frac{(x - x_0) \dots (x - x_{i-1})(x - x_{i+1}) \dots (x - x_n)}{(x_i - x_0) \dots (x_i - x_{i-1})(x_i - x_{i+1}) \dots (x_i - x_n)}$$



Запишем более кратко:

$$l_i(x) = \prod_{j=0, j \neq i}^n \frac{(x - x_j)}{(x_i - x_j)}$$

откуда

$$L_n(x) = \sum_{i=0}^n \left( \prod_{j=0, j \neq i}^n \frac{(x - x_j)}{(x_i - x_j)} \right) f(x_i)$$

Если ввести  $\omega(x)$  — угловой многочлен, который равен  $\omega_n(x) = (x - x_0) \dots (x - x_{i-1})(x - x_{i+1}) \dots (x - x_n)$ , то

$$l_i(x) = \frac{\omega_n(x)}{(x - x_i)\omega'_n(x_i)}$$

и тогда

$$L_n(x) = \sum_{i=0}^n \frac{f(x_i)\omega_n(x)}{(x - x_i)\omega'_n(x_i)}$$

**Рассчитаем погрешность полинома Лагранжа:**

Погрешность  $r_n(x) = L_n(x) - f(x)$ . Рассмотрим вспомогательную функцию  $\Delta(x) = r_n(x) - K\omega_n(x)$ .

Выясним, чему равна  $r_n(x^*)$ , где  $x^* \in [a, b]$  и  $x^* \neq x_i$ .

Потребуем, чтобы  $\Delta(x^*) = 0$ , то есть

$$K = \frac{r_n(x^*)}{\omega_n(x^*)}$$

$\Delta(x)$  имеет  $n + 2$  корня на  $[a, b]$ , следовательно, по теореме Ролля  $\Delta'(x)$  имеет  $n + 1$  корень между  $\min(x_0, x^*)$  и  $\max(x_n, x^*)$ . У  $\Delta^{(n+1)}(x)$  существует 1 корень и он в  $[a, b]$ . Пусть это некоторая точка  $\xi$ . То есть  $\Delta^{(n+1)}(\xi) = r_n^{(n+1)}(\xi) - K \cdot (n+1)! = -f^{(n+1)}(\xi) - K(n+1)! = 0$ , отсюда  $r_n(x^*) = -\frac{f^{(n+1)}(\xi)}{(n+1)!}\omega_{n+1}(x^*)$ . И теперь в силу произвольности, так как  $x^*$  — любое, то

$$r_n(x) = -\frac{f^{(n+1)}(\xi)}{(n+1)!}\omega_{n+1}(x)$$

Оценим сверху погрешность полинома Лагранжа:

$$|r_n(x)| \leq \frac{M_{n+1}}{(n+1)!} |\omega(x)|$$

где  $M_{n+1} \geq |f^{(n+1)}(x)|$ ,  $x \in [a, b]$ .

Для более гладкой  $f$ :  $M_{n+1} = \max_{x \in [a, b]} |f^{(n+1)}(x)|$ .

Рассмотрим  $f(x) = M_{n+1}x^{n+1} + P_n(x)$ , тогда  $f^{(n+1)}(x) = M_{n+1}$  и  $R_n = \frac{M_{n+1}}{(n+1)!}\omega(x)$ , то есть  $|R_n| = \frac{M_{n+1}}{(n+1)!} |\omega(x)|$ .

## 11 Кратное интерполирование. Интерполяционный полином Эрмита

Пусть на промежутке  $[a, b] \subseteq D(f)$  расположены  $m + 1$  несовпадающих узлов  $x_0, \dots, x_m$  и известны значения функции в этих точках, а также некоторые ее производные, максимальный порядок которых различен в разных узлах. Такие узлы называются **кратными**. Конкретно, считаем, что в

$x_0$	$y_0, y'_0, \dots, y_0^{(k_0-1)}$
$x_1$	$y_1, y'_1, \dots, y_1^{(k_1-1)}$
$\vdots$	$\vdots$
$x_m$	$y_m, y'_m, \dots, y_m^{(k_m-1)}$

заданы значения

тогда кратность узлов  $x_0, \dots, x_m$  соответственно равна  $k_0, \dots, k_m$ .

Предполагая, что суммарная кратность узлов есть  $\sum_{i=0}^m k_i = n + 1$ , ставим задачу построения многочлена  $H_n(x)$  степени  $n$ , такого, что  $H_n^{(j)}(x_i) = f^{(j)}(x_i) \forall i \in \overline{0, m}, \forall j \in \overline{0, k_i - 1}$  (условия Эрмитовой интерполяции). Такой многочлен будем называть **интерполяционным многочленом Эрмита**.

Формально задача состоит в определении коэффициентов многочлена  $H_n(x) = a_0x^n + a_1x^{n-1} + \dots + a_{n-1}x + a_n$ .

**Теорема 11.1.** *Многочлен Эрмита существует и единственен.*

*Доказательство.*

Единственность:

Если  $\exists H_n(x), \hat{H}_n(x)$  — два многочлена степени  $n$ , удовлетворяющие  $n + 1$  условиям Эрмитовой интерполяции, то их разность  $H_n(x) - \hat{H}_n(x)$  — многочлен степени не выше  $n$  — имеет  $n + 1$  корень с учетом кратностей, следовательно,  $H_n(x) - \hat{H}_n(x) \equiv 0$ .

Существование:

Возьмем  $f(x) = 0$ . Все ее производные равны нулю, то есть условия Эрмитовой интерполяции для нее имеют вид  $H_n^{(j)}(x_i) = 0$  и таких условий  $n + 1$ . Получается, многочлен степени  $n$  имеет  $n + 1$  корень, то есть это нуль-многочлен, то есть все его коэффициенты  $a_k$  равны нулю. С другой стороны, это однородная СЛАУ относительно коэффициентов  $a_k$ , что в силу единственности такого набора означает наличие только тривиального решения, что означает равенство нулю ее детерминанта, влекущее разрешимость такой системы при любых правых частях, то есть существование наборов таких наборов коэффициентов  $a_k$ , обеспечивающих выполнение условий эрмитовой интерполяции для любой заданной функции.  $\square$

Построим интерполяционный многочлен Эрмита:

Пусть  $L_m(x)$  — интерполяционный многочлен Лагранжа, построенный по данным  $m + 1$  значениям. Обозначим  $\Pi_{m+1}(x) = (x - x_0) \dots (x - x_m)$ . Так как по условию  $m \leq n$ , то по теореме о делении с остатком, многочлен Эрмита можно представить в виде

$$H_n(x) = L_m(x) + H_{n-(m+1)} \cdot \Pi_{m+1}(x)$$

Для построения многочлена  $H_{n-(m+1)}(x)$  привлечем информацию о производных, т.е. равенства  $H'_n(x_i) = f'(x_i)$  в тех узлах, где производные заданы. Продифференцировав равенство выше, получим

$$H'_n(x) = L'_m(x) + H'_{n-(m+1)}(x) \cdot \Pi_{m+1}(x) + H_{n-(m+1)}(x) \cdot \Pi'_{m+1}(x)$$

Поскольку  $\Pi_{m+1}(x) = 0$  в тех узлах  $x_i$ , где по условию эрмитовой интерполяции справедливо  $H'(x_i) = f'(x_i)$ , то можно записать

$$L'_m(x_i) + H_{n-(m+1)}(x_i) \cdot \Pi'_{m+1}(x) = f'(x_i)$$

откуда значение многочлена  $H_{n-(m+1)}(x)$ :

$$H_{n-(m+1)} = \frac{f'(x_i) - L'_m(x_i)}{\Pi'_{m+1}(x_i)}$$

## 12 Интерполяция сплайнами. Общие сведения. Сплайн $S_{3,2}$

В тех случаях, когда промежуток  $[a, b]$ , на котором надо интерполировать функцию  $f(x)$  функцией  $\varphi(x)$  велик и отсутствуют основания считать данную функцию  $f(x)$  достаточно гладкой, применяют кусочно-полиномиальную аппроксимацию, предполагающую, что аппроксимирующая функция  $\varphi(x)$  составляется из многочленов одинаковой небольшой степени, определяемых каждый на своей части отрезка  $[a, b]$ .

**Определение 12.1.** Сплайном  $S_m(x)$  называется определенная на  $[a, b]$  функция, принадлежащая классу  $C^l[a, b]$ , такая, что на каждом промежутке  $[x_{k-1}, x_k]$  — это многочлен  $m$ -й степени. Разность  $d = m - l$  называется дефектом сплайна.

Если сплайн  $S_m(x)$  строится по некоторой функции  $f(x)$  так, чтобы выполнялись условия  $S_m(x_i) = f(x_i)$ , то такой сплайн называется интерполяционным сплайном для функции  $f(x)$ ; при этом узлы сплайна  $x_k$  вообще говоря могут не совпадать с узлами интерполяции  $x_i$ .

**Определение 12.2.** Кубическим сплайном дефекта 1 интерполирующим на отрезке  $[a, b]$  данную функцию  $f(x)$  называется функция

$$g(x) = \{g_k(x) = a_k + b_k(x - x_k) + c_k(x - x_k)^2 + d_k(x - x_k)^3 \text{ при } x \in [x_{k-1}, x_k]\}_{k=1}^n$$

удовлетворяющая совокупности условий:

- 1)  $g(x_k) = f_k$  (условие интерполирования в узлах сплайна);
- 2)  $g(x) \in C^2[a, b]$  (двойная непрерывная дифференцируемость);
- 3)  $g''(a) = g''(b) = 0$  (краевые условия).

Для построения по данной функции  $f(x)$  интерполирующего ее сплайна  $S_{3,2}$  нужно найти  $4n$  его коэффициентов  $a_k, b_k, c_k, d_k$  ( $k = \overline{1, n}$ ). Для этого расшифруем фигурирующие в определении условия через функции  $g_k(x)$ , составляющие  $g(x)$ , имея в виду, что в любом внутреннем узле сплайна должны совпадать значения двух соседних звеньев сплайна и двух их первых производных.

Имеем:

- 1) Из условий интерполяции:

$$g_1(x_0) = f(x_0), \quad g_k(x_k) = f(x_k), \text{ при } k = \overline{1, n};$$

- 2) Из условий гладкой стыковки звеньев сплайна:

$$g_{k-1}(x_{k-1}) = g_k(x_{k-1}), \quad g'_{k-1}(x_{k-1}) = g'_k(x_{k-1}), \quad g''_{k-1}(x_{k-1}) = g''_k(x_{k-1}) \text{ при } k = \overline{2, n}.$$

- 3) Из краевых условий:

$$g''_1(x_0) = 0, \quad g''_n(x_n) = 0.$$

Как видим, коэффициентов получилось  $4n$ , то есть необходимое нам количество. Подставляя сюда выражения функций

$$g_k(x) = a_k + b_k(x - x_k) + c_k(x - x_k)^2 + d_k(x - x_k)^3$$

$$g'_k(x) = b_k + 2c_k(x - x_k) + 3d_k(x - x_k)^2$$

$$g''_k(x) = 2c_k + 6d_k(x - x_k)$$

получаем детализированную систему связей.

Теперь ставим задачу выявления эффективного способа нахождения коэффициентов сплайна. Для этого сведем систему связей к выражению всего через  $c_i$ . Обозначим  $h_i = x_i - x_{i-1}$ . В ходе магических операций, получим следующие формулы:

$$a_i = f(x_i)$$

$$h_i c_{i-1} + 2(h_i + h_{i+1})c_i + h_{i+1}c_{i+1} = 6 \left( \frac{f_{i+1} - f_i}{h_{i+1}} - \frac{f_i - f_{i-1}}{h_i} \right)$$

$$d_i = \frac{c_i - c_{i-1}}{h_i}$$

$$b_i = \frac{1}{2}h_i c_i - \frac{1}{6}h_i^2 d_i + \frac{f_i - f_{i-1}}{h_i} = \frac{f_i - f_{i-1}}{h_i} + \frac{h_i(2c_i + c_{i-1})}{6}$$

И, учитывая, что  $c_0 = c_n = 0$  находим эти значения при помощи метода прогонки, суть которого в том, чтобы сдохнуть от безнадёги.

## 13 Интерполяция сплайнами. Общие сведения. Локальные (Эрмитовы) сплайны

Предположим, что у аппроксимируемой функции известны не только значения  $f(x_i)$   $i \in \overline{0, n}$ , но и ее производные  $f'(x_i), \dots, f^{(m)}(x)$ . Как известно, по этой информации мы можем построить единственный интерполяционный многочлен Эрмита степени  $(m+1)(n+1) - 1$ . Но даже при сравнительно небольших  $m$  и  $n$  эта степень может оказаться неоправданно высокой. Здесь имеется в виду как сложность построения и использования полинома Эрмита высокой степени, так и повышенные требования к гладкости функций, для которых они служат приближениями.

Напрашивается идея сделать гибрид интерполяционного сплайна и интерполяционного многочлена. Такой гибрид назовем эрмитовыми сплайнами.

Очень легко вообразить, что по данным в точках  $x_0, x_1$  можно построить эрмитов сплайн  $H_{2m+1}^1(x)$ , по точкам  $x_1, x_2$  — эрмитов сплайн  $H_{2m+1}^2(x)$ . Так как все эти сплайны строятся независимо, то многочлены

$$S_{2m+1}(x) = \{H_{2m+1}(x), x \in [x_{i-1}, x_i]\}_{i=1}^n$$

называются также локальными сплайнами.

Свойство локальности эрмитова сплайна дает определенные преимущества при его построении, особенно в случае большого количества элементарных промежутков, а также позволяет составлять эрмитов сплайн из звеньев разных степеней, если на каких-то участках промежутка интерполяции функция  $f$  представлена производными одного порядка, а на другом — другого.



- 14 Задачи численного интегрирования. Постановка задачи. Квадратурные формулы средних прямоугольников, трапеций и Симпсона. Оценка погрешности
- 15 Интерполяционные квадратурные формулы и квадратурные формулы Ньютона-Котса. Алгоритм построения. Оценка погрешности
- 16 Квадратурные формулы наивысшей алгебраической степени точности (типа Гаусса). Алгоритм построения. Оценка погрешности
- 17 Составные квадратурные формулы. Сходимость квадратурного процесса по составным квадратурным формулам. Составная квадратурная формула средних прямоугольников
- 18 Составные квадратурные формулы на основе формул с единичным весом (средних прямоугольников, трапеций, Симпсона). Оценка погрешности и алгебраическая степень точности
- 19 Практические способы оценки погрешности численного интегрирования составными квадратурными формулами (правило Рунге). Процесс Эйткена для оценки порядка аппроксимации
- 20 Методы типа Рунге-Кутты решения начальной задачи для ОДУ. Порядок и число этапов. Оценка методической погрешности на шаге
- 21 Практические способы оценки погрешности методов типа Рунге-Кутты. Вложенные методы
- 22 Алгоритм решения ОДУ методом типа Рунге-Кутты с автоматическим выбором шага