

# ЧИСЛЕННЫЕ МЕТОДЫ МАТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ.

## (ЛЕКЦИИ темы:0-3)

### Тема 0: Основные понятия курса.

#### П.1. Характеристики алгоритмов численных методов.

1. **Погрешность** (точность вычисления);
2. **Трудоемкость** (необходимое число действий, вычислений);
3. **Необходимая память.**

#### П.2. Абсолютная и относительная погрешности.

Пусть  $x$  - приближенное значение некоторой величины, а  $x_0$  - ее точное значение, тогда абсолютной погрешностью  $x$  называется величина  $\Delta x = |x - x_0|$ , тогда:  $x = x_0 \pm \Delta x$

Относительной погрешностью измерения называется величина :

$$\delta x = \frac{\Delta x}{|x_0|} \quad x = x_0 (1 \pm \delta x)$$

Обязательное условие:  $\Delta x \ll |x|$ ,  $\delta x \ll 1$

В значении погрешности необходимо учитывать, приблизительно, до 1,2 значащих знаков.

Значение величины берется на 1-2 знака больше величины погрешности.

Пример:

$$X = 9,86543 \pm 0,01395$$

$$9,865 \pm 0,01$$

$$98654 \pm 0,014$$

$$X = 3,7867 \pm 0,578$$

$$3,79 \pm 0,6$$

$$3,787 \pm 0,58$$

#### П.3. Изменение абсолютной и относительной погрешностей в результате арифметических операций.

##### Теорема 1.1.

При сложении абсолютные погрешности складываются, т.е. абсолютная погрешность суммы не превосходит сумму абсолютных погрешностей.

$$x = x_0 \pm \Delta x$$

$$y = y_0 \pm \Delta y$$

$$x + y = x_0 \pm \Delta x + y_0 \pm \Delta y = (x_0 + y_0) \pm (\Delta x \pm \Delta y)$$

Абсолютная погрешность разности не превосходит сумму абсолютных погрешностей.

**Задача:** Исследовать изменение относительной погрешности при сложении и вычитании. (8 баллов)

*Замечание:*

При многократном суммировании величины с абсолютной погрешностью  $\Delta x$  итоговая абсолютная погрешность будет не  $n \cdot \Delta x$ , а несколько меньше - порядка  $\sqrt{n} \Delta x$  (по закону больших чисел).

### **Теорема 1.2.**

При умножении и делении складываются относительные погрешности, т.е. относительная погрешность произведения (частного) не превосходит сумму относительных погрешностей.

$$x = x_0(1 \pm \delta x)$$

$$y = y_0(1 \pm \delta y)$$

$$xy = x_0 y_0 (1 \pm \delta x)(1 \pm \delta y) = x_0 y_0 (1 \pm \delta x \pm \delta y \pm \delta x \delta y)$$

$$\delta x \delta y \ll \delta x, \delta y$$

Задача: доказать утверждение для деления (5 баллов).

Задача: умножение и деление. Использовать изменение абсолютной погрешности при умножении и делении (6 баллов).

### **П.4. Изменение погрешности при вычислении функций.**

Абсолютная погрешность при вычислении функции равна:

$$\begin{aligned} f(x) = f(x_0 \pm \Delta x) &\approx f(x_0) \pm f'(x_0) \Delta x = f(x_0 \pm \Delta x) - f(x_0) = \\ &= f(x_0) + \frac{(f(x_0 \pm \Delta x) - f(x_0)) \Delta x}{\Delta x} = f'(x_0) \Delta x \end{aligned}$$

1. Степенная функция:

$$f(x) = x^\alpha$$

$$f(x) = f(x_0 \pm \Delta x) = (x_0 \pm \Delta x)^\alpha = (x_0(1 \pm \delta x))^\alpha = x_0^\alpha (1 \pm \delta x)^\alpha \approx x_0^\alpha (1 \pm \alpha \delta x)$$

Вывод: при вычислении степенной функции относительная погрешность увеличивается в  $|\alpha|$  раз.

2. Экспоненциальная функция:

$$f(x) = e^x \quad x = x_0 + \Delta x$$

$$e^x = e^{x_0 \pm \Delta x} = e^{x_0} e^{\pm \Delta x} = e^{x_0} (1 \pm \Delta x)$$

Вывод: относительная погрешность при вычислении экспоненты равна абсолютной погрешности аргумента.

3. Логарифмическая функция:

$$\ln x = \ln x_0 \pm \frac{1}{x_0} \Delta x = \ln x_0 \pm \delta x$$

Вывод: абсолютная погрешность при вычислении логарифмической функции равна относительной погрешности аргумента.

Правило работы с погрешностями:

1. Вычисления погрешностей производить с точностью до 0.01;
2. Значение величины берется на 1-2 знака больше порядка погрешности.

Задача: исследовать относительную, абсолютную погрешности при вычислении **sin** (7 баллов).

После этого все элементы первого столбца под главной диагональю будут нулевыми, а  $a_{11}$  – ведущий элемент на первом этапе. На втором этапе

обнуляются все элементы второго столбца под главной диагональю, т.е. вторая строка умножается на  $(-a_{32}/a_{22})$  и суммируется с третьей и т.д.. Необходимое количество этапов -  $(n-1)$ . В результате, первоначальная СЛАУ (1,1) будет приведена к СЛАУ в форме верхней треугольной матрицы, которая легко решается "обратным ходом" (порядок вычисления:  $x_n$ , затем  $x_{n-1}, \dots, x_2, x_1$ ).

Пример: рассмотрим систему:

$$\begin{cases} -x - y + z = -1 \\ 2x - y + 2z = 3 \\ 3x - y + z = 3 \end{cases} \quad (1.3)$$

$$\left( \begin{array}{ccc|c} -1 & -1 & 1 & -1 \\ 2 & -1 & 2 & 3 \\ 3 & -1 & 1 & 3 \end{array} \right) \rightarrow \left( \begin{array}{ccc|c} -1 & -1 & 1 & -1 \\ 0 & -3 & 4 & 1 \\ 0 & -4 & 4 & 0 \end{array} \right) \rightarrow \left( \begin{array}{ccc|c} -1 & -1 & 1 & -1 \\ 0 & -3 & 4 & 1 \\ 0 & 0 & -\frac{4}{3} & -\frac{4}{3} \end{array} \right), \text{ на первом этапе обнуляются}$$

2-ая и 3-я строки в 1-ом столбце, на втором 3-я строка во 2-ом столбце, на этом прямой ход закончен:  $z=1, y=1, x=1$ .

Модификация метода Гаусса – метод Гаусса с выбором наибольшего элемента.

В ходе решения системы (1,3) при обнулении первого столбца ведущим элементом был:  $a_{11}=-1$ , а ведущим элементом при обнулении второго столбца был:  $a_{22}=-3$ .

Отличие метода Гаусса с выбором наибольшего элемента от обычного метода Гаусса состоит в том, что на каждом шаге в качестве ведущего элемента выбирается наибольший по модулю среди возможных (т.е. стоящих не выше главной диагонали). После этого осуществляется перестановка строк, чтобы наибольший ведущий элемент стоял на главной диагонали. Остальные этапы решения аналогичны обычному методу Гаусса.

Решение системы (1,3) с помощью метода Гаусса с выбором наибольшего элемента

$$\left( \begin{array}{ccc|c} -1 & -1 & 1 & -1 \\ 2 & -1 & 2 & 3 \\ 3 & -1 & 1 & 3 \end{array} \right) \rightarrow \left( \begin{array}{ccc|c} 3 & -1 & 1 & 3 \\ 2 & -1 & 2 & 3 \\ -1 & -1 & 1 & -1 \end{array} \right) \rightarrow \text{обнуляется 1 столбец.}$$

Рассмотрим первый столбец, выбираем максимальный элемент и меняем местами строки

$$\rightarrow \left( \begin{array}{ccc|c} 3 & -1 & 1 & 3 \\ 0 & -1/3 & 4/3 & 1 \\ 0 & -4/3 & 4/3 & 0 \end{array} \right) \rightarrow \left( \begin{array}{ccc|c} 3 & -1 & 1 & 3 \\ 0 & -4/3 & 4/3 & 0 \\ 0 & -1/3 & 4/3 & 1 \end{array} \right) \rightarrow \left( \begin{array}{ccc|c} 3 & -1 & 1 & 3 \\ 0 & -4/3 & 4/3 & 0 \\ 0 & 0 & 1 & 1 \end{array} \right), z=1, y=1, x=1$$

Замечание: Метод Гаусса с выбором наибольшего элемента работает заметно точнее обычного метода Гаусса и позволяет решить на ЭВМ систему с большим количеством АЛУ.

Оценим трудоемкость метода Гаусса:

Система уравнений  $n \times n$ . Необходимо обнулить  $(n-1)$  столбец. Это потребует  $3(n+1)[(n-1)+(n-2)+\dots+1]$  действий при прямом ходе.  $(n-1)$ -число обнуляемых

строк в первом столбце;  $(n-2)$  – число обнуляемых строк во втором столбце;  $3$  – число операций сложения строк).

При обратном ходе:  $1+3+5+\dots+(2n-1)$ : количество действий равно  $n^2$  ( $n$  – количество действий при вычислении  $x_n$ ).

Всего действий:

$$\frac{3(n+1)(n-1)n}{2} + n^2 \approx \frac{3}{2}n^3$$

При модифицированном методе Гаусса потребуется еще  $n^2$  операций сравнения и  $n^2$  перестановок элементов.

## **П.2. Метод квадратных корней (МКК)- точный метод решения СЛАУ.**

Метод квадратных корней применим для решения СЛАУ с положительно определенной симметричной матрицей.

Определение: матрица  $A$  называется симметричной, если  $A=A^T$ , т.е.  $a_{ij}=a_{ji}$ .

Основная идея метода К.К.: представить исходную СЛАУ  $Ax=b$  к виду,

$T^T * Tx=b$ , для этого ищется верхняя треугольная матрица  $T$  – такая, что  $T^T * T=A$ .

Решение системы (1,4)  $T^T * Tx=b$  проводится в два этапа:

1. находится  $y$  – такое, что  $T^T * y=b$  ( $T^T$  – нижняя треугольная матрица) – это осуществляется аналогично обратному ходу метода Гаусса, так как  $T^T$  – нижняя треугольная матрица.
2. Находится  $x$  – такое, что  $Tx=y$ , оно и будет искомым.

Основная задача: разложить матрицу  $A=T^T * T$  (1,5).

Неизвестными являются коэффициенты матрицы:

$$T = \begin{pmatrix} t_{11} & t_{12} & \dots & t_{1n} \\ 0 & t_{22} & \dots & t_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & t_{nn} \end{pmatrix}, \Rightarrow T^T = \begin{pmatrix} t_{11} & 0 & \dots & 0 \\ t_{12} & t_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ t_{1n} & t_{2n} & \dots & t_{nn} \end{pmatrix}$$

$$T^T * T = A$$

$$a_{ij} = t_{1i} * t_{1j} + t_{2i} * t_{2j} + \dots + t_{ni} * t_{nj} = t_{1i} * t_{1j} + t_{2i} * t_{2j} + \dots + t_{ii} * t_{ij} + 0 + \dots + 0, \text{ где } i \leq j$$

$$\text{находим } t_{11}, \Rightarrow a_{11} = t_{11}t_{11} \rightarrow t_{11} = \sqrt{a_{11}},$$

$$\text{находим } t_{12}, \Rightarrow a_{12} = t_{11}t_{12} \rightarrow t_{12} = \frac{a_{12}}{t_{11}},$$

$$j=2, \dots, n \quad t_{1j} = \frac{a_{1j}}{t_{11}}, \text{ м.к.: } a_{1j} = t_{1j}t_{11}$$

$$\text{аналогично: } t_{2j} = \frac{a_{2j} - t_{12}t_{1j}}{t_{22}}; j=3, \dots, n$$

Получаем следующий набор формул для МКК:

$$t_{11} = \sqrt{a_{11}} \quad t_{1j} = \frac{a_{1j}}{t_{11}} \quad (j=2, 3, \dots, n)$$

$$t_{ii} = \sqrt{a_{ii} - \sum_{k=1}^{i-1} t_{ki} t_{ki}} \quad t_{ij} = \frac{a_{ij} - \sum_{k=1}^{i-1} t_{ki} t_{kj}}{t_{ii}}; \text{при } i > j, t_{ij} = 0$$

Метод КК применим в чистом виде только для СЛАУ с симметричной положительно определенной матрицей. Любую СЛАУ можно привести к подобной с помощью метода симметризации. Для этого исходную СЛАУ умножают на матрицу  $A^T$  слева (обе части  $Ax=b$ ) при этом получают  $A^T A x = A^T b$ , после чего решают систему  $Cx=d$ , где  $C=A^T A$ ,  $d=A^T b$ . В этой системе матрица  $C$  – симметричная и положительно определенная, т.е. для нее применим МКК, полученная система эквивалентна исходной.

## **§2. Итерационные (приближенные) методы решения СЛАУ.**

### **П.1. Справочный материал.**

Пусть  $x=(x_1, x_2, \dots, x_n)$  –  $n$  – мерный вектор.

Его нормой называется – число  $\|x\|$ ,

$$\|x\| = \|(x_1, x_2, \dots, x_n)\|$$

Удовлетворяющее условиям:

1.  $\|x\| \geq 0$ , если  $\|x\|=0$ , то  $x=0=(0, 0, \dots, 0)$
2.  $\|\alpha x\| = |\alpha| \cdot \|x\|$
3.  $\|x+y\| \leq \|x\| + \|y\|$

Примеры норм:

1. Евклидова норма ( $l_2$ -норма):

$$\|(x_1, x_2, \dots, x_n)\|_2 = \sqrt{x_1^2 + x_2^2 + \dots + x_n^2}$$

2. Первая норма ( $l_1$ -норма):

$$\|(x_1, x_2, \dots, x_n)\|_1 = |x_1| + |x_2| + \dots + |x_n|$$

3. Бесконечная норма ( $l_\infty$ -норма):

$$\|(x_1, x_2, \dots, x_n)\|_\infty = \max(|x_1|, |x_2|, \dots, |x_n|)$$

Пример:

$$X=(1, 2, -2)$$

$$\|(1, 2, -2)\|_2 = 3$$

$$\|(1, 2, -2)\|_1 = 5$$

$$\|(1, 2, -2)\|_\infty = 2$$

**Определение:**

Нормой матрицы  $A$  называется число :

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

### **Теорема 1:**

Норма матрицы удовлетворяет всем свойствам нормы

Доказательство:

1.  $\|A\| \geq 0$  – очевидно, т.к.  $\frac{\|Ax\|}{\|x\|} \geq 0$

Если  $\|A\|=0$ , то  $\|A_x\|=0$  для всех  $x$  и т.к.  $A_x$  – вектор, то  $A_x=0$  (для всех  $x$ ), но это возможно только тогда, когда все координаты матрицы  $A$   $a_{ij}=0$ .

Если у матрицы  $A$  имеется  $a_{ij} \neq 0$ , то взяв  $x=(0,0,1,0,0)$  получим  $A_x = \begin{pmatrix} a_{1j} \\ a_{2j} \\ \vdots \\ a_{nj} \end{pmatrix}$  -  $i$ -тое

место. (остальное доказать по 3 бала).

### **Теорема 1.2:**

$$\|AB\| \leq \|A\| * \|B\| \quad (1.6).$$

Доказательство:

Имеем:

$$\|A\| = \max_{\|x\| \neq 0} \frac{\|A_x\|}{\|x\|} = \max_{\|x\| \neq 0} \frac{\|AB_x\|}{\|B_x\| \|x\|} \leq \max_{\|x\| \neq 0} \frac{\|AB_x\|}{\|B_x\|} = \max_{\|x\| \neq 0} \frac{\|B_x\|}{\|B_x\|} = \max_{\|y\| \neq 0} \frac{\|A_y\|}{\|y\|} * \max_{\|x\| \neq 0} \frac{\|B_x\|}{\|x\|} = \|A\| * \|B\|$$

Примеры матричных норм:

$$\|A\|_1 = \max_{\|x\| \neq 0} \frac{\|A_x\|_1}{\|x\|_1} \quad \text{тах среди сумм по столбцам}$$

$$\|A\|_{2,\infty} = \max_{\|x\| \neq 0} \frac{\|A_x\|_{2(\infty)}}{\|x\|_{2(\infty)}} \quad \text{тах среди сумм по строкам}$$

### **Теорема 1.3:**

$$A=n*n\text{- матрица.} \quad \|A\|_1 = \max_j \sum_{i=1}^n |a_{ij}| \quad \|A\|_\infty = \max_i \sum_{j=1}^m |a_{ij}|$$

Доказательство:

$$\begin{aligned} \|A\|_1 &= \max_{x \neq 0} \frac{\|A_x\|_1}{\|x\|_1} = \max_{x \neq 0} \frac{\sum_{i=1}^n |A_x|_i}{\sum_{j=1}^m |x_j|} = \max_{x \neq 0} \frac{\sum_{j=1}^m \sum_{i=1}^n |a_{ij} x_j|}{\sum_{j=1}^m |x_j|} \leq \max_{x \neq 0} \frac{\sum_{i=1}^n \sum_{j=1}^m |a_{ij} x_j|}{\sum_{j=1}^m |x_j|} = \\ &= \max_{x \neq 0} \frac{\sum_{j=1}^m \sum_{i=1}^n |a_{ij} x_j|}{\sum_{j=1}^m |x_j|} = \max_{x \neq 0} \frac{\sum_{j=1}^m |x_j| \sum_{i=1}^n |a_{ij}|}{\sum_{j=1}^m |x_j|} \leq \max_{x \neq 0} \frac{\sum_{j=1}^m |x_j| \max_j \sum_{i=1}^n |a_{ij}|}{\sum_{j=1}^m |x_j|} = \max_j \sum_{i=1}^n |a_{ij}| \end{aligned}$$

Доказали для первой нормы.

Необходимо доказать, что в неравенстве достигаются равенства.

## **П.2. Метод простой итерации.**

Рассмотрим СЛАУ  $Ax=b$  с квадратной невырожденной матрицей  $A=n*n$ .

Предположим все диагональные элементы матрицы  $A$  не равны нулю. Тогда поделив все строки СЛАУ на соответствующие диагональные  $a_{ij}$ , получим матрицу, у которой диагональными элементами являются единицы.

Матрица А имеет вид :

$$A = \begin{pmatrix} 1 & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \dots & 1 \end{pmatrix}, \text{ представим матрицу } A \text{ как сумму матриц: } A=E+C$$

$$E = \begin{pmatrix} 1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 1 \end{pmatrix} \quad C = \begin{pmatrix} 0 & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \dots & 0 \end{pmatrix}$$

Получаем:  $(E+C)x=v$

$$Ex+Cx=v$$

$$x+Cx=v$$

$$x=v-Cx \quad (1.7)$$

(1.7)- эквивалентно первоначальной СЛАУ (1.2).

Рассмотрим итерационный процесс:

$$x^{(k+1)}=v-Cx^{(k)} \quad (1.8).$$

#### **Теорема 1.4:**

Если итерационный процесс сходится (т.е. существует вектор:  $x^{(\infty)} = \lim_{k \rightarrow \infty} x^{(k)}$ , то предельный вектор  $x^{(\infty)}$  этого итерационного процесса будет решением (1.7), а следовательно и исходной системы (1.2).

Доказательство:

Перейдем в (1.8) к пределу при  $k \rightarrow \infty$  получим:  $\lim_{k \rightarrow \infty} x^{(k+1)} = \lim_{k \rightarrow \infty} v - Cx^{(k)} \Rightarrow$

$$x^{\infty}=v-Cx^{(\infty)}, \text{ ч.т.д.}$$

Исследуем условия, при которых итерационный процесс сходится.

#### **Замечание:**

Для того, чтобы запустить итерационный процесс необходимо задать значения первоначального приближения  $x^{(0)}$ , а все остальные значения  $x^{(k)}$  вычисляются по формуле (1.8) .

В качестве  $x^{(0)}$  обычно берут 0, в этом случае  $x^{(1)}=v$  или  $x^{(0)}=v$ .

#### **Теорема 1.5** (достаточное условие сходимости итерационного процесса (1.8)):

Если  $\|C\|<1$  (любая норма 1,2,...,∞), то процесс (1.8) сходится.

Доказательство:

Считаем, что  $(k_1 > k_2)$ , тогда:  $x^{(k_1)} - x^{(k_2)} = (\sum_{i=k_2+1}^{k_1} C^i (-1)^i) b$  и

$$\begin{aligned} \|x^{(k_1)} - x^{(k_2)}\| &= \left\| \left( \sum_{i=k_2+1}^{k_1} C^i (-1)^i \right) b \right\| \leq \sum_{i=k_2+1}^{k_1} \|C^i (-1)^i b\| \leq \sum_{i=k_2+1}^{k_1} \|C^i\| * \|b\| \leq \sum_{i=k_2+1}^{k_1} \|C\|^i * \|b\| = \\ &= \frac{\|C\|^{k_1+1} - \|C\|^{k_2+1}}{\|C\| - 1} * \|b\| \rightarrow 0 \end{aligned}$$

итерационный процесс (1.8) необходимо продолжать до тех пор, пока погрешность решения  $\|x^k - x^{\infty}\|$  не будет меньше  $\varepsilon$  - заданной погрешности. Иногда оценивают не погрешность решения, а погрешность первой части  $(Ax^{(k)} - Ax^{(\infty)})$



Оценим погрешность решения через погрешность правой части и наоборот:

$Ax=v$ ,  $x$  - точное решение, найдено  $\bar{x}$  - приближенное решение  $\Rightarrow A\bar{x}=\bar{v}$ . Если  $A\bar{x}=\bar{v}$ , тогда  $\Delta x=x-\bar{x}$  - погрешность решения,  $\Delta v=v-\bar{v}$  - погрешность правой части.

$$\underbrace{A(x-\bar{x})}_{=A\Delta x} = Ax - A\bar{x} = \underbrace{v-\bar{v}}_{=\Delta v} \quad (1.9).$$

$\|\Delta b\|=\|A*\Delta x\|\leq\|A\|*\|\Delta x\|$  - оценка погрешности правой части через погрешность решения.

Умножив (1.9) на  $A^{-1}$  слева, получим:

$$A*A^{-1}\Delta x=A^{-1}\Delta b$$

$$\Delta x=A^{-1}\Delta b$$

$\|\Delta x\|\leq\|A^{-1}\|*\|\Delta b\|$  - оценка погрешности решения через погрешность правой части.

Число  $C_A=\|A\|*\|A^{-1}\|$  называется коэффициентом обусловленности матрицы. Матрица, у которой  $C_A\gg 1$ , называется плохо обусловленной. При решении СЛАУ с такими матрицами погрешности сильно возрастают.

### **П.3. Оценка скорости сходимости метода простой итерации.**

Формула итерационного процесса:

$$x^{(k)} = (E + \sum_{i=1}^k C^i (-1)^i) b \quad (1.10),$$

1. если  $\|C\|<1$ , то (как доказано в теореме (1.5)) процесс сходится.

$$x = x^{(\infty)} = \lim x^{(k)} = (E + \sum_{i=1}^{\infty} C^i (-1)^i) b \quad (1.11)$$

Вычисляя разность двух уравнений (1.11)-(1.10), получаем следующее соотношение-

$$x - x^{(k)} = (\sum_{i=k+1}^{\infty} C^i (-1)^i) b$$

$$\|x - x^{(k)}\| = \left\| \left( \sum_{i=k+1}^{\infty} C^i (-1)^i \right) b \right\| \leq \left\| \sum_{i=k+1}^{\infty} C^i (-1)^i \right\| \|b\| \leq \sum_{i=k+1}^{\infty} \|C^i\| * \|b\| = \frac{\|C\|^{k+1}}{1 - \|C\|} * \|b\|$$

- обладающее свойствами геометрической прогрессии со знаменателем  $\|C\|$ .

$\|\Delta b\|\leq\|A\|*\|\Delta x\|\leq\|A\|*(\|C\|^{k+1}/(1-\|C\|))*\|b\|$  - норма оценки правой части.

Условие диагонального преобладания:

Условие -  $\|C\|<1$  – равносильно тому, что для исходной матрицы системы (до деления ее на диагональные элементы) выполняется условие диагонального преобладания по строкам.

$\overline{a_{ij}}$  - элементы первоначальной матрицы

2.  $\sum_{i \neq j} \left| \frac{a_{ij}}{a_{ii}} \right| < 1$ ; т.е.:  $\sum_{i \neq j} |a_{ij}| < |a_{ii}|$  - условие диагонального преобладания по строкам, т.е.

элемент, стоящий на диагонали, больше суммы модулей всех остальных элементов своей строки. Это является достаточным условием сходимости процесса простой итерации.

**Задача (6 баллов):** используя норму с индексом 1 получить достаточное условие сходимости процессов простой итерации.

3.  $\sum_{i \neq j} |a_{ij}| < |a_{ii}|$  - условие диагонального преобладания по столбцам.

#### **П.4. Метод Зейделя.**

Исходную СЛАУ  $Ax=b$  тем же приемом, что и в методе простой итерации, приведем к виду  $x=b-Cx^{(k)}$  :

$$\begin{cases} x_1^{(k+1)} = b_1 - (0 + a_{12}x_2^k + a_{13}x_3^k + \dots + a_{1n}x_n^k \\ x_2^{(k+1)} = b_2 - (a_{21}x_1^k + 0 + a_{23}x_3^k + \dots + a_{2n}x_n^k \\ \dots \dots \dots \\ x_n^{(k+1)} = b_n - (a_{n1}x_1^k + a_{n2}x_2^k + \dots + 0 \end{cases} \quad (1.12)$$

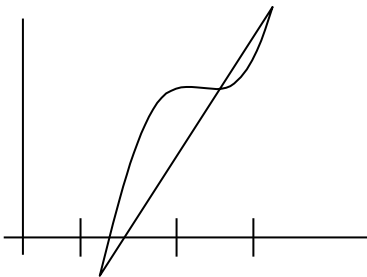
При нахождении  $x_i^{(k+1)}$  (на  $k+1$  шаге) используется  $x_1^{(k+1)}, x_2^{(k+1)}, \dots, x_{i-1}^{(k+1)}$  найденные на шаге  $(k+1)$  и  $x_{i+1}^{(k)}, \dots, x_n^{(k)}$  - найденные на предыдущем шаге итерации.

Метод Зейделя работает несколько быстрее метода простой итерации. Скорость сходимости метода сравнима со скоростью сходимости процесса простой итерации. Условие (любое из трех) диагонального преобладания будет достаточным для его сходимости.

## **Тема 2. Методы решения нелинейных уравнений и системы нелинейных уравнений (НУ и СНУ).**

### **§1. Метод половинного деления.**

Рассмотрим  $f(x)=0$ : нелинейное уравнение, где  $f(x)$  – некоторая нелинейная функция (непрерывная).



### Алгоритм метода:

1. Найти интервал  $[a, b]$ , на котором функция  $f$  меняет свой знак, т.е.  $f(a) \cdot f(b) < 0$ .
2. Поделить интервал  $[a, b]$  пополам точкой  $c = (a+b)/2$ .
3. Выяснить, на какой из половин  $[a, c]$  или  $[c, b]$  функция  $f$  меняет знак, и в качестве нового отрезка рассмотреть ту половину, на которой происходит смена знака. После чего повторить пункт (2) (если решение не было найдено с заданной точностью  $\varepsilon$ . Если же оно было найдено, то (4))
4. Ответ:  $x = (a+b)/2 \pm ((b-a)/2 < \varepsilon)$  – погрешность решения.

Скорость сходимости метода половинного деления: погрешность на каждом шаге итерации уменьшается вдвое, т.е. для добавления одного верного десятичного знака надо проделать 3-4 (шага) итерации.

Точность:  $\frac{b-a}{2}$ , где  $b$  и  $a$  – левая и правая границы последнего полученного

интервала. Необходимое число шагов:  $N = \left\lceil \log_2 \frac{b-a}{\varepsilon} \right\rceil$

### **§2. Метод Хорд – модификация метода половинного деления.**

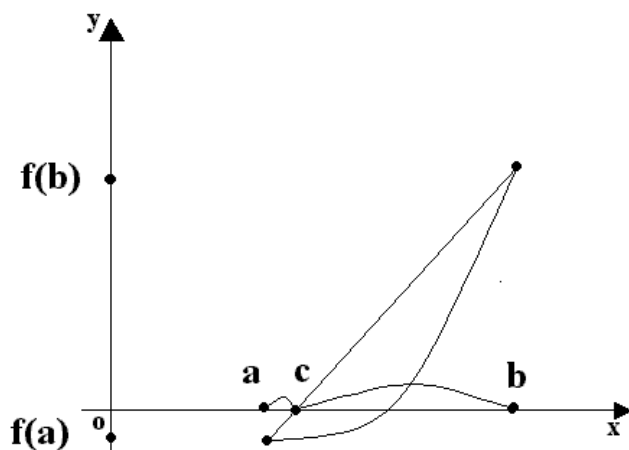
В МПД интервал многократно делится пополам, а в методе хорд интервал делится на неравные части, пропорциональные  $f(a)$  и  $f(b)$ , т.е. точка  $c$  – это точка пересечения хорд с осью  $OX$ .

#### Алгоритм метода:

1. Найти интервал  $[a, b]$ , такой, что  $f(a) \cdot f(b) < 0$ .
2. Найти соединение этих точек хордой.

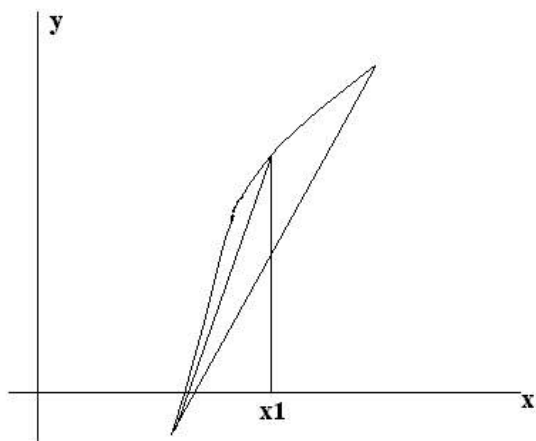
Найти пересечение хорды с осью  $OX$   
(т.е.  $x_1 = \frac{f(b) \cdot a - f(a) \cdot b}{f(b) - f(a)}$ )

3. Из отрезков  $[a, x_1]$  и  $[x_1, b]$  выбрать тот, на котором функция меняет знак. Если заданная точность не достигнута, то повторить пункт (2).



#### Критерий прерывания итерационного процесса:

$|C^{(k)} - C^{(k+1)}| < \varepsilon$  – если данное условие выполнено, то процесс прекращается, а в качестве точного значения берется  $C^{(k+1)}$ . Данный критерий прерывания рекомендуется использовать во всех итерационных процессах, если явно не указан какой-либо другой теоретический обоснованный критерий.



### Оценка скорости сходимости в методе хорд.

#### **Теорема 2.1:**

Если на интервале  $[a,b]$  функция  $f$  непрерывна и дифференцируема, а ее производная  $f'$  на интервале  $[a,b]$  имеет постоянный знак (т.е. на  $[a,b]$   $f$  – монотонна), то верна следующая оценка:

$|x - x_n| \leq ((M-m)/m) * |x_n - x_{n-1}|$ , где:

$$m = \min_{x \in [a,b]} |f'(x)|, \quad M = \max_{x \in [a,b]} |f'(x)|, \quad x \in [a,b].$$

#### **Следствие 2.2:**

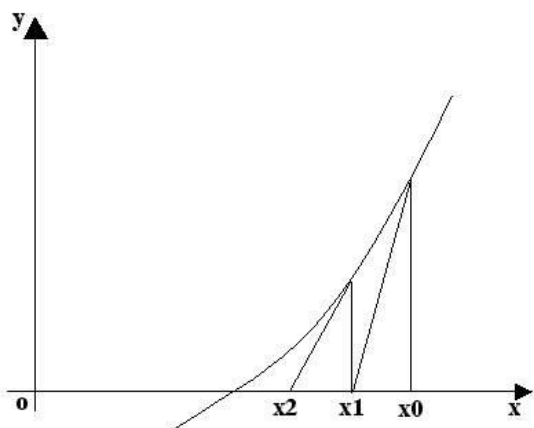
Если  $M$  близко по своему значению к  $m$  (например:  $M \leq 2m$ ), то  $|x - x_n| \leq |x_n - x_{n-1}|$ , следовательно, как только разность между двумя последовательно приближенными  $x_n$  и  $x_{n-1}$  меньше  $\varepsilon$ , т.е.  $|x_n - x_{n-1}| < \varepsilon$ , то  $x_n$  можно взять в качестве искомого решения. Это универсальный критерий прерывания.

#### **Следствие 2.3:**

Скорость сходимости в методе хорд не хуже геометрической прогрессии со знаменателем  $(M-m)/m$ , т.е.  $|x - x_n| \leq ((M-m)/m)^n * (b-a)$ , т.е. в целом метод хорд сходится быстрее метода половинного деления, если  $(M-m)/m < 1/2$

## **§ 3 Метод Ньютона (метод касательных)**

### **п1. Идея и алгоритм метода:**



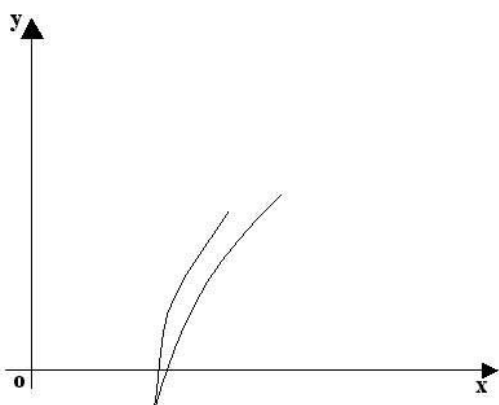
1. В качестве первоначального приближения  $x_0$  выбрать любую точку, достаточно близкую к корню.
2. В точке  $(x_0)$  провести касательную к графику функции до пересечения с осью  $OX$  в точке  $x_1$ .
3. В точке  $(x_1, f(x_1))$  провести касательную до пересечения с осью  $OX$ , получить  $x_2$  и т.д.

Повторяем процесс пока не достигнута заданная точность.

$x_n, f(x_n)$   $k=f'(x_n)$ ,  $k$  – угловой коэффициент

$$y=f(x_n)+k(x-x_n)=f(x_n)+f'(x_n)(x-x_n)=0$$

$$x=x_n - (f(x_n)/f'(x_n)), \text{ т.е. } x_{n+1}=x_n - (f(x_n)/f'(x_n))$$



### **Теорема 2.4**

Если функция  $f(x)$  дважды непрерывна и дифференцируема на  $[a,b]$ ,  $f'$  и  $f''$  на  $[a,b]$  не меняют своих знаков (т.е. функция монотонна и выпукла вверх или вниз), то имеет место следующее неравенство:

$$|x-x_{n+1}| \leq 1/2(M_2/m_1) * |x-x_n|^2, \text{ где}$$

$$m_1 = \min_{x \in [a,b]} |f'(x)|$$

$$M_2 = \max_{x \in [a,b]} |f''(x)| \text{ обеспечивает эффект удвоения знаков после каждой итерации.}$$

Перепишем (2.1) в матричном виде:

$$\begin{aligned} f_1(x_1, x_2, \dots, x_n) &= f_1(x_1^0, x_2^0, \dots, x_n^0) \\ f_2(x_1, x_2, \dots, x_n) &= f_2(x_1^0, x_2^0, \dots, x_n^0) \\ &\vdots \\ f_n(x_1, x_2, \dots, x_n) &= f_n(x_1^0, x_2^0, \dots, x_n^0) \end{aligned} + \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \dots & \frac{\partial f_n}{\partial x_n} \end{pmatrix}_{x_0} \begin{pmatrix} x_1 - x_1^0 \\ x_2 - x_2^0 \\ \vdots \\ x_n - x_n^0 \end{pmatrix}$$

Все частные производные существуют в точке  $x_0$ . Заменим в (2.1)  $\approx$  на  $=$  ... и найдем  $x$ .

$$F(x_0) + \partial F|_{x_0}(x - x_0) = 0$$

$$F(x_0) + W(x - x_0) = 0$$

$$W^{-1} * W(x - x_0) = -F(x_0)$$

$$W^{-1} * W(x - x_0) = W^{-1}F(x_0)$$

$$\text{получим: } x = x_0 - W^{-1}F(x_0)$$

Найденный  $x$  возьмем в качестве следующего приближения:  $x_{n+1} = x_n - W^{-1}F(x_n)$ , где  $W^{-1}_n$  - матрица, обратная  $W_n$ , т.е.  $W_n = \partial F|_{x_n}$  - матрица частных производных в точке  $x_n$ .

Многомерный метод Ньютона, так же как и метод Ньютона работает быстро (удваивая знаки после каждой итерации), но не всегда срабатывает (если не удачно выбрано  $x_0$ ).

Итерационный процесс продолжать до тех пор, пока разность  $|x_n - x_{n-1}|$  не станет меньше  $\varepsilon$ . Как только это произошло, в качестве решения взять  $x_n$ .

Замечание:

Для нахождения  $Y = W^{-1}F$  не обязательно находить матрицу  $W^{-1}$ , так как  $Y$  - решение системы линейных уравнений ( $W$  - известно,  $F$  - известно).

#### § 4. Метод итерации

Рассмотрим нелинейные уравнения вида  $x = U(x)$  (2.2) и итерационный процесс  $x_{n+1} = V(x_n)$  (2.3).  $U$  - непрерывная и гладкая.

#### Теорема 2.5

Если итерационный процесс сходится, а  $U$  непрерывна в  $x^\infty$ , то предельное значение  $x^\infty = \lim x_n$  будет решением исходного НУ.

Доказательство:

$$\text{Перейдем к (2.3) при } n \rightarrow \infty. \lim x_{n+1} = \lim (V(x_n)) = V(\lim (x_n)) = U(x_\infty)$$

Замечание:

Итерационный процесс (2.3) не всегда сходится.

Пример:

$$x^2 - 2 = 0$$

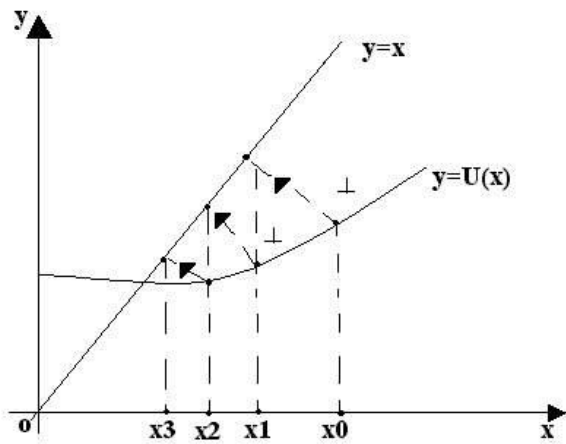
$$\text{приведем к виду } x = U(x)$$

$$x = x^2 + x - 2$$

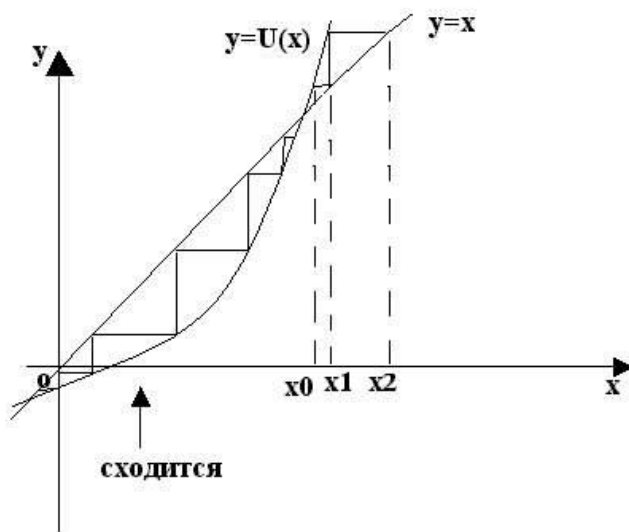
возьмем в качестве  $x_0 = 1$  получим:  $x_1 = 0$ ,  $x_2 = -2$ ,  $x_3 = 0$ ,  $x_4 = -2$ . Значит, последовательность итерации не сходится, и метод не работал.

Попытаемся разобраться, когда итерационный процесс сходится.

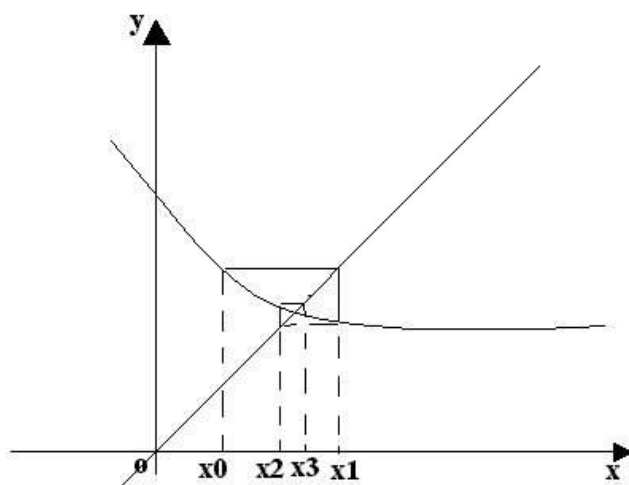
Рассмотрим график:



В данном случае итерационный процесс сходится к точному решению -  $x_\infty$ .

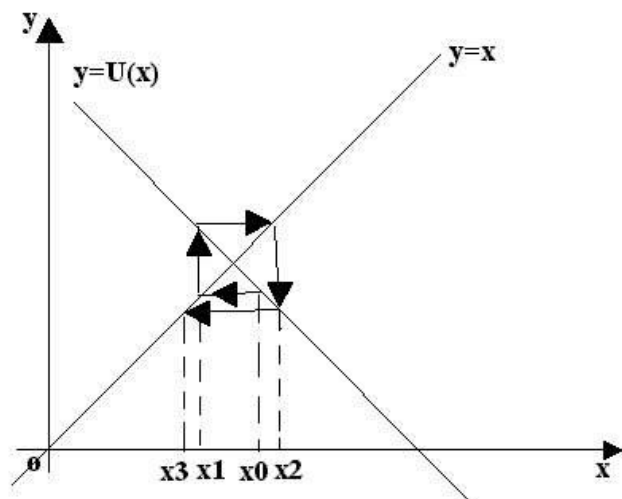


В данном случае итерационный процесс расходится.



Итерационный процесс сходится, но не монотонно.





Итерационный процесс расходится (по спирали).

Вывод: итерационный процесс сходится, если  $|U'(x)| < 1$  вблизи  $x_\infty$ .

### Многомерный вариант метода итерации.

Рассмотрим систему нелинейных уравнений (СНУ)

$$(2.4) \quad x = U(x), \text{ где } x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}, \quad U = \begin{pmatrix} U_1(x_1, \dots, x_n) \\ \vdots \\ U_n(x_1, \dots, x_n) \end{pmatrix}$$

тогда итерационный процесс  $x_{n+1} = U(x_n)$  (2.5) сходится или расходится, в зависимости от выполнения условий функции  $U$  и ее производной. Если он сходится, то его предельное значение  $x_\infty$  - решение уравнений (2.4).

### Доказательство:

Такое же, как в теореме 2.5.

### Теорема 2.6 (с сжимающим отображением)

Пусть  $U$  - вектор функция. Из  $|\mathbb{R}^n| \rightarrow \mathbb{R}^n$  (т.е. у нее  $n$ - аргументов и  $n$  координат). Тогда, если  $V$  отображает некоторую область  $G$  в себя (в  $G$ ) и является сжимающим отображением, т.е. существует  $g < 1$ , такое, что  $\|U(X) - U(Y)\| \leq g \|X - Y\|$ , то итерационный процесс (2.5) сходится, следовательно, существует неподвижная точка  $x_\infty$  (т.к.  $x_\infty = U(x_\infty)$ ) при этом скорость сходимости итерационного процесса (2.5) - аналогична геометрической прогрессии со знаменателем  $g$ , т.е.

$$\|x_n - x_\infty\| \leq \text{const } g^n.$$

### Доказательство:

Заметим, что  $\|x - x_{n+1}\| = \|U(x_n) - V(x_{n-1})\| \leq$  (т.к.  $U$  - сжимающая отображение)  $\leq q \|x - x_{n+1}\|$  (2.6), тогда (пусть  $q > 1$ ) имеем  $\|x_k - x_1\| \leq \|(x_k - x_{k-1}) + (x_{k-1} - x_{k-2}) + \dots + (x_{1+1} - x_1)\| \leq$  (по неравенству треугольников)  $\leq \|(x_k - x_{k-1})\| + \|(x_{k-1} - x_{k-2})\| + \dots + \|(x_{1+1} - x_1)\| \leq$  (2.7)  $\leq \|(x_1 - x_0)\| q^{k-1} + \|(x_1 - x_0)\| q^{k-2} + \dots + \|(x_1 - x_0)\| q^1 \rightarrow 0$  (2.8). - последовательность Коши.

Следовательно, итерационный процесс сходится.

На основании (2.6) получаем:  $\|(x_{n-1} - x_n)\| \leq \|(x_n - x_{n-1})\| q \leq \|(x_1 - x_0)\| q^n$  (2.7).

Для того, чтобы доказать, что скорость сходимости – геометрическая прогрессия, оценим скорость сходимости итерационного процесса. Для этого подставим в формулу (2.8)  $k = \infty$ ,  $l = n$ , получим :

$$\|x_{\infty} - x_n\| \leq \frac{q^{\infty} - q^n}{q - 1} \|x_1 - x_0\| = q^n \frac{\|x_1 - x_0\|}{1 - q} = \text{const}$$

### **Теорема 2.7**

Если  $U$  отображение  $U$  из  $|R^n \rightarrow R^n|$  матрица частных производных

$$W = dU = \begin{pmatrix} \frac{\partial U_1}{\partial x_1} & \cdots & \frac{\partial U_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial U_n}{\partial x_1} & \cdots & \frac{\partial U_n}{\partial x_n} \end{pmatrix} \text{ имеет норму всюду } \leq q, \text{ то отображение } U \text{ сжимающее}$$

с коэффициентом  $U$ , т.е.  $\|U(X) - V(Y)\| \leq q \|X - Y\|$

Доказательство: (15 баллов)

Если сопоставить теорему 2.6 и теорему 2.7, то получаем, что итерационный процесс 2.5 сходится со скоростью геометрической прогрессии со знаменателем  $q$ , если норма матрицы частных производных  $\|W\| \leq q$ .

## **Тема 3: Интерполирование**

### **§ 0. Постановка задач интерполяции.**

#### **Основная идея.**

#### **Задача:**

Пусть имеются точки  $x_0, x_1, x_2, \dots, x_n$  в которых известны значения функции  $f$  – соответственно  $y_0, y_1, y_2, \dots, y_n$ .

Необходимо вычислить функцию  $f$  в любой другой заданной точке  $x$ . Если  $x$  принадлежит  $x \in [x_0, x_n]$ , то это задача интерполяции. Если  $x$  не принадлежит  $x \notin [x_0, x_n]$ , то это задача экстраполяции.

**Основная идея интерполяции:** заменить функцию  $f$  на некоторую другую функцию  $y(x)$  (которая выбирается из некоторого фиксированного класса функций. Например, многочленов, сплайнов, тригонометрических функций), такую что  $y(x_i) = y_i$ ,  $i = \overline{0, n}$  (т.е. любое от 0 до  $n$ ).

Конкретизируем эту процедуру.

Пусть  $M$  – некоторый заданный класс функций, в котором мы будем искать нужную нам функцию  $g$ .

Класс  $M$  всякий раз подбирается некоторым специальным образом, в зависимости от того, какую функцию  $f$  мы интерполируем, но при этом класс  $M$  должен быть достаточно широким, чтобы интерполирующая функция обладала требуемыми свойствами, и не очень широким, чтобы интерполирующая функция  $g$  была единственной.

В зависимости от  $M$  – класса интерполирующих функций, мы будем рассматривать 3 вида интерполяции:

- интерполяция полиномами;
- интерполяция сплайнами;
- интерполяция тригонометрическими функциями.

**§ 1. Интерполяция многочленами**  
**(т.е. функция выбирается среди многочлена)**  
**п 1. Интерполяционный многочлен Лагранжа.**

**Теорема 3.1**

Для любых  $x_0 < x_1 < x_2 \dots < x_n$  и  $y_0, y_1, y_2 \dots y_n$  существует единственный многочлен  $p \in P_n$  (т.е. многочлен  $p$  в степени  $\leq n$ ) такой, что  $p(x_i) = y_i, i = \overline{0, n}$

Доказательство:

Докажем сначала единственность многочлена  $p$ . Предположим, что существует два интерполирующих многочлена  $p_1$  и  $p_2$ . Имеем  $P_1(x_i) = y_i$

$P_2(x_i) = y_i$ , где  $i = \overline{0, n}$ . Рассмотрим многочлен  $h = P_1 - P_2$ , очевидно его степень не выше  $n$ .  $h(x_i) = 0$  имеет не менее  $(n+1)$  корней. Как известно из алгебры, у ненулевого многочлена степени  $n$  корней не может быть больше  $n$ , следовательно,  $h$  тождественно 0.  $h \equiv 0$  и  $P_1 = P_2$

Докажем теперь существование многочлена  $p$ : рассмотрим для этого следующий набор многочленов

$$P(x) = \sum_{i=0}^n y_i q_i(x), \text{ где } q_i(x) = \frac{(x-x_0)(x-x_1)\dots(x-x_{i-1})(x-x_{i+1})\dots(x-x_n)}{(x_i-x_0)(x_i-x_1)\dots(x_i-x_{i-1})(x_i-x_{i+1})\dots(x_i-x_n)}$$

Заметим, что все  $q_i$  многочлены степени  $n$ , следовательно,  $P_n(x)$  будет многочлен степени не выше  $n$ . Докажем, что  $p_n(x)$  искомый, т.е.  $p_n(x_i) = y_i$  для этого подсчитаем  $q_i$  в точках  $x_j$

$$q_i(x_j) = \begin{cases} 0, & j \neq i \\ 1, & j = i \end{cases}$$

Следовательно,  $p(x) = y_0 q_0(x) + \dots + y_j q_j(x) + \dots + y_n q_n(x) = y_j$

Таким образом, многочлен построен. Он называется интерполяционным многочленом Лагранжа.

Как следует из теоремы 3.1, такой интерполяционный многочлен единственен, но формул для него существует много, все эти формулы дают один и тот же результат.

Пример:

$n=1$  (интерполируем по двум точкам)

$$p_1(x) = y_0 \frac{x-x_1}{x_0-x_1} + y_1 \frac{x-x_0}{x_1-x_0}$$

$n=2$  (интерполируем по трем точкам)

$$p_2(x) = y_0 \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} + y_1 \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} + y_2 \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}$$

## Схема Эйткена в вычислении интерполяционного многочлена.

**Теорема 3.2:** если  $P_{x_0, x_1 \dots x_{n-1}}(x)$  - многочлен, интерполирующий функцию  $f$  в точках  $x_0 \dots x_{n-1}$  (степени не выше  $n-1$ ), а  $P_{x_1, x_2 \dots x_n}(x)$  - многочлен интерполирующий функцию в точках  $x_1 \dots x_n$  (степени не выше  $n-1$ ), то многочлен  $P_{x_0, x_1 \dots x_n}(x)$  - многочлен интерполирующий функцию в точках  $x_0 \dots x_n$  (степени не выше  $n$ ) может быть вычислена по формуле:

$$P_{x_0, x_1 \dots x_n}(x) = \frac{P_{x_0, x_1 \dots x_{n-1}}(x)(x - x_n) - P_{x_1, x_2 \dots x_n}(x)(x - x_0)}{x_0 - x_n}$$

Доказательство:

Многочлен  $P_{x_0, x_1 \dots x_n}(x)$  имеет степень не выше  $n$ .

Осталось проверить, что данный многочлен в узлах интерполяции задает значения  $y_i$ .  $P_{x_0, x_1 \dots x_n}(x_i) = y_i, i = \overline{0, n}$

Рассмотрим три возможности:

1.  $i=0$

$$P_{x_0, x_1 \dots x_n}(x_0) = \frac{P_{x_0, x_1 \dots x_{n-1}}(x_0)(x_0 - x_n) - P_{x_1, x_2 \dots x_n}(x_0)(x_0 - x_0)}{x_0 - x_n} = \frac{y_0(x_0 - x_n) - P_{x_1, x_2 \dots x_n}(x_0) * 0}{x_0 - x_n} = y_0$$

2.  $i=n$

$$P_{x_0, x_1 \dots x_n}(x_n) = \frac{P_{x_0, x_1 \dots x_{n-1}}(x_n)(x_n - x_n) - P_{x_1, x_2 \dots x_n}(x_n)(x_n - x_0)}{x_0 - x_n} = y_n$$

3.  $i = \overline{0, n-1}$

$$P_{x_0, x_1 \dots x_n}(x_i) = \frac{P_{x_0, x_1 \dots x_{n-1}}(x_i)(x_i - x_n) - P_{x_1, x_2 \dots x_n}(x_i)(x_i - x_0)}{x_0 - x_n} = y_i$$

### Формула Эйткена для слияния двух интерполяционных многочленов

$$P_{x_k, x_{k+1} \dots x_l}(x) = \frac{P_{x_k, x_{k+1} \dots x_l}(x)(x - x_l) - P_{x_{k+1}, x_{k+2} \dots x_l}(x)(x - x_k)}{x_k - x_l}$$

### Трудоёмкость метода Эйткена.

Трудоёмкость одного слияния – 7.

Всего слияний  $n + (n-1) + (n-2) + \dots + 1 = \frac{n(n+1)}{2}$

$$\frac{7n(n+1)}{2} \sim \frac{7}{2}n^2$$

У схемы Эйткена, по сравнению с формулой Лагранжа, имеется два больших преимущества:

1. более устойчива по сравнению с формулой Лагранжа к вычислительным погрешностям (возникающих при округлении на вычислительной машине).
2. в отличие от формулы Лагранжа, обрывается, т.е. вычисления по схеме Эйткена можно не доводить до конца и при этом будет получено значение близкое к точному.

Объясняется это тем, что если в схеме Эйткена вычисления до конца не довести, то будет получен интерполяционный многочлен, который интерполирует функцию не во всех узлах  $x_0 \dots x_n$ , а только лишь в некоторых из них. При этом его значение близко к значению интерполяционного многочлена  $P_{x_0 \dots x_n}$ .

Очень часто при вычислении по схеме Эйткена поступают следующим образом: добавляют по одному узлу интерполяции, пока не будет достигнута заданная точность.

#### Замечание:

При практической реализации схемы Эйткена выгодно не добавлять точки интерполяции с одной стороны (слева или справа), а добавлять их поочередно – одну справа, одну слева.

## **п 2. Оценка погрешности интерполяционного многочлена Лагранжа.**

При интерполировании возникает два типа погрешностей:

1. погрешность усечения (возникает из-за замены функции на интерполирующий многочлен);
2. погрешность округления (возникает из-за того, что значения интерполируемой функции  $f$  в узлах интерполяции известны не точно, а приближенно, с некоторой погрешностью  $\eta$ )

Очевидно, что  $\varepsilon_{\text{реальное}}$  не превосходит  $\varepsilon_y + \varepsilon_0$

Если узлы -  $x_0 \dots x_n$  - равноотстоящие, то  $\varepsilon_0 \leq 2^{n+1} \eta$

Для  $\varepsilon_{\text{усеч}}$  справедлива оценка:

$$\varepsilon_{\text{усеч}} = \frac{M_{n+1}}{(n+1)!} |(x-x_0)(x-x_1)\dots(x-x_n)|$$

$$M_{n+1} = \max |f^{(n+1)}(t)|, \text{ где } t \in [x_0, x_n, x]$$

**Теорема:**  $\varepsilon_{\text{усеч}}$  с учетом знака для интерполирующего многочлена

$$r_n(x) = f(x) - P_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} (x-x_0)\dots(x-x_n), \text{ где } \xi \text{ некоторая точка } \in [x_0, x_n, x]$$

#### Доказательство:

Рассмотрим  $\Pi(x) = (x-x_0)\dots(x-x_n)$  со старшим коэффициентом равным 1. Введем функцию  $U(x) = r_n(x) - k\Pi(x)$ , где  $k$  некоторая const подобранная специальным образом, для этого фиксируем точку  $\bar{x}$ , не совпадающую ни с одним узлом интерполяции

$$k = \frac{r_n(\bar{x})}{\Pi(\bar{x})}$$

$$U(x) = f(x) - P_n(x) - \frac{r_n(\bar{x})}{\Pi(\bar{x})} \Pi(x)$$

$$U(x_i) = f(x_i) - P_n(x_i) - k * 0 = 0 \quad i = \overline{0, n}$$

$$U(\bar{x}) = r_n(\bar{x}) - \frac{r_n(\bar{x})}{\Pi(\bar{x})} \Pi(\bar{x}) = 0$$

Следовательно, функция  $U$  на интервале  $[x_0, x_n, x]$  обращается в 0, как минимум  $(n+2)$  раза. Тогда, ее производная  $U'$  обращается в 0, как минимум  $(n+1)$  раз.  $U''$  как минимум  $n$  раз. Следовательно,  $U^{(n+1)}$  обращается на этом интервале хотя бы один раз в 0, т.е. существует  $\xi \in [x_0, x_n, \bar{x}]$

$$U^{(n+1)}(\xi) = 0$$

$$U^{(n+1)}(\xi) = f^{(n+1)}(\xi) - 0 - k(n+1)! = 0$$

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

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

Заменим  $\bar{x}$  на  $x$  и получим формулу.

Пример вычисления интерполяционного многочлена и оценка  $\varepsilon_{\text{усеч}}$  в узлах  $x_0=100, x_1=121, x_2=144, y_0=10, y_1=11, y_2=12$ . Найдем  $\sqrt{115}$ , используя интерполяцию по трем точкам.

$$P_2(115) = 10 \frac{(115-121)(115-144)}{(100-121)(100-144)} + 11 \frac{(115-100)(115-144)}{(121-100)(121-144)} + 12 \frac{(115-121)(115-100)}{(144-100)(144-121)} = 10.7228$$

$$\varepsilon_{\text{реальное}} = 1 \cdot 10^{-3}$$

$$\text{Оценим } \varepsilon_{\text{усеч}} \varepsilon_{\text{усеч}} \leq \frac{M_3}{3!} |(115-100)(115-121)(115-144)| = 2 \cdot 10^{-3}$$

$$f''' = \frac{3}{8} x^{-\frac{5}{2}}$$

$$M_3 = \max \left| \frac{3}{8} x^{-5/2} \right| = \frac{3}{8} 10^{-5}$$

$\varepsilon_{\text{окр}}=0$ , т.к. значения функции в узлах интерполяции были известны точно.

$$\varepsilon_{\text{реальное}} = 10^{-3} \leq \varepsilon_{\text{усеч}} + \varepsilon_{\text{окр}} = 2 \cdot 10^{-3}$$

### **п 3. Конечные разности интерполяционный многочлен Ньютона**

Конечной разностью функции  $y=f(x)$  называется функция  $\Delta y(x) = f(x+h) - f(x)$ , где  $h$  — фиксированный шаг. Конечные разности иногда называются конечными разностями первого порядка.

При взятии конечной разности от многочлена  $n$ -ой степени, его степень понижается на единицу, а старший коэффициент увеличивается в  $(n \cdot h)$  раз.

$$\Delta^{n+1} y(x) = \Delta(\Delta^n y(x))$$

Пример:

$$\begin{aligned} \Delta^2 y(x) &= \Delta(\Delta y(x)) = \Delta(f(x+h) - f(x)) = f(x+2h) - f(x+h) - [f(x+h) - f(x)] = \\ &= f(x+2h) - 2f(x+h) + f(x) \end{aligned}$$

(похоже на бином Ньютона)

Конечная разность от многочлена.

Пусть  $y(x) = P_n(x) = a_0 x^n + a_1 x^{n-1} + \dots + a_n$

$$\Delta y = a_n (x+h)^n + a_{n-1}(x+h)^{n-1} + \dots + a_0 - [a_n x^n + a_{n-1} x^{n-1} + \dots + a_0] = [\text{бином Ньютона}] =$$

$$= a_n ([x^n + c_n^1 x^{n-1} h + c_n^2 x^{n-2} h^2 + \dots] - x^n) + a_{n-1} ([x^{n-1} + c_{n-1}^1 x^{n-2} h + c_{n-1}^2 x^{n-3} h^2 + \dots] - x^{n-1}) + \dots =$$

$$= a_n h x^{n-1} + \dots x^{n-2} + \dots$$

Конечная разность похожа на производную.

### **Теорема 3.2**

$$\Delta^n f(x) = h^n f^{(n)}(\xi), \quad \text{где точка } \xi \in [x_0, x_n]$$

(формула справедлива, если функция дифференцируема n раз)

Следствие:

При n=1 данная теорема называется теоремой Лагранжа.

$$\Delta f(x) = hf'(\xi)$$

Если за  $\Delta$  обозначить тождественный оператор, то после применения оператора получим:  $(1+\Delta)f(x) = f(x) + \Delta f(x) = f(x) + f(x+h) - f(x) = f(x+h)$ . Таким образом, оператор  $(1+\Delta)f$  – оператор превращения и сдвига на h (увеличивает аргумент на h). Соответственно, если оператор применить n раз, то:

$$(1+\Delta)^n f(x) = f(x+nh)$$

Свойство оператора  $\Delta$ :

1.  $\Delta(f+g) = \Delta f + \Delta g$
2.  $\Delta(\alpha f) = \alpha \Delta f$
3.  $\Delta(fg) = \Delta f \cdot g + f \cdot \Delta g$
4.  $\Delta^k(\Delta^l f) = \Delta^{k+l} f$

Если функция  $f(x)$  задана своими значениями  $y_i$  в равноотстоящих узлах  $x_i$  с шагом  $h$ ,  $x_i = x_0 + ih$ ,  $i = \overline{0, h}$ , то конечные разности в точках  $x_i$  удобно вычислять с помощью таблицы конечных разностей.

Рассмотрим функцию  $f(x) = 2x^3 - 2x^2 + 3x - 1$

$$x_i = x_0 + ih = 0 + i \cdot 1, \quad i = \overline{0, 5}$$

x	y	$\Delta y$	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$	$\Delta^5 y$
0	-1	3	8	12	0	0
1	2	11	20	12	0	
2	13	31	32	12		
3	44	63	44			
4	107	107				
5	214					

### **Обобщенная степень и ее свойства.**

Обобщенной степенью числа  $x$  называется выражение

$$x^{[n]} = x(x-h)(x-2h)\dots(x-(n-1)h) \quad (\text{шаг } h \text{ фиксированная константа}), \quad x^{[0]} = 0, \quad x^{[1]} = x.$$

При  $h \rightarrow 0$  обобщенная степень превращается в обычную степень.

Вычислим конечную разность обобщенной степени.

$$\Delta^k x^{[n]} = x^{[n-k]} n(n-1)\dots(n-k+1)h^k$$

Построение интерполяционного многочлена Ньютона (первая интерполяционная формула Ньютона)

$$\text{Строим многочлен } P_n(x) = a_0 + a_1(x-x_0) + a_2(x-x_0)(x-x_1) + \dots + a_n(x-x_0)(x-x_1)\dots(x-x_{n-1})$$

Задача: найти неизвестные коэффициенты многочлена  $a_0, a_1, \dots, a_n$ .

Найдем сначала  $a_0$ , имеем  $P_n(x_i) = y_i$ ,  $y_0 = P_n(x_0) = a_0 + 0 + \dots \Rightarrow a_0 = y_0$ . Найдем  $a_1$ , для этого считаем первую конечную разность в точке  $x_0$ . Для этого, многочлен  $P_n(x)$  перепишем через обобщенные степени:

$$P_n(x) = a_0 + a_1(x-x_0) + a_2(x-x_0)(x-x_0-h) + a_3(x-x_0)(x-x_0-h)(x-x_0-2h) + \dots + a_n(x-x_0)(x-x_0-h) \dots (x-x_0-(n-1)h) = a_0 + a_1(x-x_0)^{[1]} + a_2(x-x_0)^{[2]} + \dots + a_n(x-x_0)^{[n]}$$

Используя, свойства конечных разностей получаем

$$\Delta P_n(x) = \Delta a_0 + a_1 * 1 * h(x-x_0)^{[0]} + a_2 * 2h(x-x_0)^{[1]} + \dots + a_n * nh(x-x_0)^{[n-1]}$$

Подставим вместо  $x - x_0$

$$\Delta y_0 = \Delta P_n(x_0) = a_1 h * 1 + a_2 * 2 * (x_0 - x_0) + 0 + \dots$$

$$a_1 = \frac{\Delta y_0}{h}$$

Для нахождения  $a_2$  составим вторую конечную разность

$$\Delta^2 y_0 = \Delta^2 P_n(x_0) = a_2 * 2 * 1 * h^2 * 1 + 0 + \dots$$

$$a_2 = \frac{\Delta^2 y_0}{2h^2}$$

Продолжая эту процедуру, приходим к выводу, что

$$\Delta^k P_n(x) = a_k * k(k-1) \dots 1 * h^k (x-x_0)^{[0]} + a_{k+1}(k+1)k \dots 2h^2(x-x_0)^{[1]} + \dots$$

$$\Delta^k y_0 = \Delta^k P_n(x_0) = a_k * k! * h^k \Rightarrow a_k = \Delta^k y_0 / (h^k * k!)$$

Окончательно:

$$\begin{aligned} P_n(x) &= a_0 + a_1(x-x_0)^{[1]} + a_2(x-x_0)^{[2]} + \dots + a_n(x-x_0)^{[n]} = \\ &= \sum_{k=0}^n a_k(x-x_0)^{[k]} = \sum_{k=0}^n \frac{\Delta^k y_0}{k! h^k} (x-x_0)^{[k]} = \sum_{k=0}^n \frac{\Delta^k y_0}{k!} q(q-1)(q-2) \dots (q-k+1) = \\ &= y_0 + \frac{\Delta y_0}{1!} q + \frac{\Delta^2 y_0}{2!} q(q-1) + \dots + \frac{\Delta^n y_0}{n!} q(q-1) \dots (q-n+1), \end{aligned}$$

где  $q = \frac{x-x_0}{h}$ ,  $\frac{(x-x_0)^{[k]}}{h^k} = q(q-1)(q-2) \dots (q-k+1)$  (3.7) — первая

интерполяционная формула Ньютона.

### Вторая интерполяционная формула Ньютона.

Она выводится аналогично первой интерполяционной формуле Ньютона и имеет вид:

$$P_n(x) = \sum_{k=0}^n \frac{\Delta^k y_{n-k}}{k!} q(q+1) \dots (q+k-1) = y_n + \frac{\Delta y_{n-1}}{1!} q + \frac{\Delta^2 y_{n-2}}{2!} q(q+1) + \dots + \frac{\Delta^n y_0}{n!} q(q+1) \dots (q+n-1),$$

$$\text{где } q = \frac{x-x_n}{h}$$

(3.8)

Сравнение интерполяционных многочленов Ньютона и Лагранжа (ИМЛ и ИМН):

Достоинство ИМЛ в том, что он работает всегда, а не только для равностоящих узлов интерполяции (как многочлен Ньютона).

Достоинство ИМН: суммирование в формулах 3.7 и 3.8 можно прерывать раньше времени. Если в этих формулах были взяты  $L$  первых слагаемых, то при этом получился интерполяционный многочлен степени  $\leq (L-1)$ , которая



интерполирует заданную функцию в  $L$  крайних узлах ( $L$  первых узлов  $x_0, x_1, \dots, x_{L-1}$  для формулы 3.7, для  $L$  последних узлов  $x_{n-L+1}, x_{n-L+2}, \dots, x_n$  для формулы 3.8) В интерполяционной формуле Лагранжа прерывать суммирование раньше времени нельзя.

Первая интерполяционная формула Ньютона применяется, когда  $x$  близко к  $x_0$ . Вторая интерполяционная формула Ньютона применяется, когда  $x$  близко к  $x_n$ .

Замечание:

Если в формулах 3.7, 3.8 оставить все слагаемые, то получится один и тот же интерполяционный многочлен, который, согласно теореме 3.1, единственен и совпадает с интерполяционным многочленом Лагранжа.

#### Погрешности 1-ого и 2-ого интерполяционных многочлена Ньютона.

Как известно (3.3) погрешность усечения при интерполировании многочлена:

$$\varepsilon_{\text{усеч}} \leq \frac{M_{n+1}}{(n+1)!} |(x-x_0)(x-x_1)\dots(x-x_n)| \approx \frac{\Delta^{n+1} y_0}{(n+1)!} |q(q-1)\dots(q-n)|, \quad \text{где } q = \frac{x-x_0}{h} \quad (3.9)$$

При использовании 1-ой и 2-ой интерполяционных формул Ньютона, возникают погрешности округления. Если  $y_i$  были известны с точностью  $\eta$ , то  $\Delta y_i$  будет известно с точностью  $-\varepsilon_{\text{округл}} \leq 2^{n-1} \eta$ .

#### п 4. Интерполяционная формула Стирлинга и Бесселя (без вывода).

Формулы Ньютона - первая и вторая - односторонние, т.е. при добавлении нового слагаемого добавляется один узел интерполяции: слева направо для первой формулы Ньютона (левосторонняя формула), справа налево для второй формулы Ньютона (правосторонняя формула). Лучше добавлять узлы интерполяции поочередно: слева и справа, тогда  $\varepsilon_{\text{усеч}}$  будет меньше.

Центральными интерполяционными формулами являются формулы Бесселя и Стирлинга.

##### Формула Стирлинга.

$$\begin{aligned} P_{2n}(x) = & y_0 + \frac{q}{1!} \frac{\Delta y_{-1} + \Delta y_0}{2} + \frac{q^2}{2!} \Delta^2 y_{-1} + \frac{q(q^2-1^2)}{3!} \frac{\Delta^3 y_{-2} + \Delta^3 y_{-1}}{2} + \frac{q^2(q^2-1^2)}{4!} \Delta^4 y_{-2} + \\ & + \frac{q(q^2-1^2)(q^2-2^2)}{5!} \frac{\Delta^5 y_{-3} + \Delta^5 y_{-2}}{2} + \frac{q(q^2-1^2)(q^2-2^2)}{6!} \Delta^6 y_{-3} + \dots + \\ & + \frac{q(q^2-1^2)\dots(q^2-(n-1)^2)}{(2n-1)!} \Delta^{2n} y_{-n}, \quad \text{где } q = \frac{x-x_0}{h} \end{aligned}$$

Это многочлен степени  $2n$ , интерполирующий функцию в узлах  $x_{-n}, x_{-n+1}, \dots, x_0, x_1, \dots, x_n$ . Формула Стирлинга допускает обрывание раньше времени. При добавлении новой пары, слагаемые добавляются по одному узлу интерполяции слева и справа.

##### Формула Бесселя.

$$\begin{aligned}
P(x) = & \frac{y_0 + y_1}{2} + p\Delta y_0 + \frac{p^2 - (1/2)^2}{2!} \frac{\Delta^2 y_{-1} + \Delta^2 y_0}{2} + \frac{p(p^2 - (1/2)^2)}{3!} \Delta^3 y_{-1} + \\
& + \frac{(p^2 - (1/2)^2)(p^2 - (3/2)^2)}{4!} \frac{\Delta^4 y_{-2} + \Delta^4 y_{-1}}{2} + \frac{p(p^2 - (1/2)^2)(p^2 - (3/2)^2)}{5!} \Delta^5 y_{-2} + \\
& + \frac{(p^2 - (1/2)^2)(p^2 - (3/2)^2)(p^2 - (5/2)^2)}{6!} \frac{\Delta^6 y_{-3} + \Delta^6 y_{-2}}{2} + \frac{p(p^2 - (1/2)^2)(p^2 - (3/2)^2)(p^2 - (5/2)^2)}{7!} \Delta^7 y_{-3} + \\
& + \dots + \frac{(p^2 - (1/2)^2)(p^2 - (3/2)^2) \dots (p^2 - ((2n-1)/2)^2)}{2n!} \frac{\Delta^{2n} y_{-n} + \Delta^{2n} y_{-n+1}}{2} + \\
& + \frac{p(p^2 - (1/2)^2)(p^2 - (3/2)^2) \dots (p^2 - ((2n-1)/2)^2)}{(2n+1)!} \Delta^{2n+1} y_{-n} \\
P = & \left( x - \frac{x_0 + x_1}{2} \right) / h
\end{aligned}$$

Суммирование в формуле Бесселя можно обрывать. При добавлении пары слагаемых добавляются две точки интерполяции: одна слева и одна справа. Преимущество этих многочленов перед ИМЛ и ИМН: большая точность при практических вычислениях.

## **§ 2. Тригонометрическая интерполяция. Интерполяция по ортонормированным системам функций.**

### **п 1. Справочный материал**

Договоренность: пусть  $x_i$  некоторая фиксированная сетка. Далее будем отождествлять функцию  $y=f(x)$  с ее набором значений в узлах  $x_i$ .

$$f \sim (f(x_0), f(x_1) \dots f(x_n))$$

и использовать комплексные числа вида:  $a+iv$ .

Скалярное произведение. Его свойства. Примеры скалярных произведений.

Скалярное произведение двух векторов  $x$  и  $y$  – число обозначаемое  $(x, y)$  и удовлетворяющее следующим свойствам.

1.  $(x, x) \geq 0$ , если  $(x, x) = 0$ , то  $x=0$
2. дистрибутивность по первому аргументу  
 $(x+y, z) = (x, z) + (y, z)$
3. однородность по первому аргументу  
 $(\alpha x, y) = \alpha(x, y)$
4.  $(y, x) = \overline{(x, y)}$

Примеры скалярного произведения:

$x=(x_0, \dots, x_n)$ ,  $y=(y_0, \dots, y_n)$  (координаты векторов)

$$(x, y) = \sum_{k=0}^n x_k \overline{y_k}$$

Система векторов  $x^{(0)} \dots x^{(m)}$  называется ортогональной, если  $k \neq l \Rightarrow (x^{(k)}, x^{(l)}) = 0$

Система векторов  $x^{(0)} \dots x^{(m)}$  называется ортонормированной, если она

1. ортогональная:  $k \neq l \Rightarrow (x^{(k)}, x^{(l)}) = 0$
2. нормированная:  $(x^{(k)}, x^{(k)}) = 1$

Примечание: обычные вектора  $i, j, k$  в трехмерном пространстве образуют ортонормированную систему.

### Теорема 3.3

Любая ортонормированная система векторов линейно независима

Доказательство:

Предположим, что найдется набор чисел  $\alpha_k$ , где  $\alpha = \overline{0, m}$

$$\sum_{k=0}^m \alpha_k x^{(k)} = 0$$

Тогда для любого  $l$ :

$$(0, x^{(l)}) = \left( \sum_{k=0}^m \alpha_k x^{(k)}, x^{(l)} \right) = \sum_{k=0}^m (\alpha_k x^{(k)}, x^{(l)}) = \sum_{k=0}^m \alpha_k (x^{(k)}, x^{(l)}) = \alpha_0 * 0 + \dots + \alpha_l * 1 + 0 \dots = \alpha_l$$

Все коэффициенты  $\alpha_l = 0$  в линейной комбинации, - получено противоречие предположению о нетривиальной линейной комбинации и доказывает теорему.

Следствие 3.4:

Ортонормированная система из  $n+1$  векторов ( $n+1$  – размерность пространства) будет базисом этого пространства

Доказательство:

По теореме 3.3 наша система линейно независима, а т.к. там ровно  $n$  векторов, то они составляют базис, в данном случае размерность пространства не  $n$ , а  $n+1$ , что и доказывает теорему.

Лемма 3.5: если  $x^{(0)} \dots x^{(n)}$  – ортонормированная система и базис пространства, тогда вектор  $(x, x^{(l)}) = 0 \quad \forall l = \overline{0, n} \Leftrightarrow x = 0$

Доказательство:

Т.к. количество векторов совпадает с размерностью пространства и они ортонормированны, то по следствию 3.4 вектора  $x_l \quad l = \overline{0, n}$

Данный вектор  $x$  разложим по этому базису:

$x = \sum_{l=0}^n \alpha_l x^{(l)}$  и данное выражение подставим в скалярное произведение

$$0 = \left( \sum_{l=0}^n \alpha_l x^{(l)}, x^{(k)} \right) = 0 + \alpha_k + 0 + 0 \dots$$

$$\text{все } \alpha_k = 0 \Rightarrow x = 0$$

$$x = \sum_{l=0}^n \alpha_l x^{(l)}$$

Теорема 3.6: (дискретное разложение Фурье). Если вектора  $x^{(0)} \dots x^{(k)}$  ортонормированны (составляют базис  $n+1$  мерного пространства), то как известно из линейной алгебры, любой вектор  $x$  можно разложить по этому базису.

$$x = \sum_{k=0}^n (x, x^{(k)}) x^{(k)}$$

Доказательство:

чтобы доказать теорему, необходимо и достаточно доказать, что вектор

$y = x - \sum_{k=0}^n (x, x^{(k)}) x^{(k)} = 0$ . используя, выше доказанную формулу приходим к

выводу, что необходимо доказать:  $(y, x^{(l)}) = 0$

$$\begin{aligned} (y, x^{(l)}) &= \left( \left[ x - \sum_{k=0}^n (x, x^{(k)}) x^{(k)} \right], x^{(l)} \right) = (x, x^{(l)}) - \sum_{k=0}^n ((x, x^{(k)}) x^{(k)}, x^{(l)}) = (x, x^{(l)}) - \sum_{k=0}^n (x, x^{(k)}) (x^{(k)}, x^{(l)}) = \\ &= (x, x^{(l)}) - (0 + \dots + 0 + (x, x^{(l)}) * 1 + 0 \dots) = 0 \end{aligned}$$

## п 2. Тригонометрическая интерполяция

### Теорема 3.7:

Система функций  $y^{(j)}(x) = \exp(2\pi i(j_0 + j)x)$ , где  $j = \overline{0, n-1}$ , а  $j_0$  - целое число, образует ортонормированную систему на сетке  $x_0 \dots x_{n-1} \dots x_k = 0 + n/k$ , где  $k = \overline{0, n-1}$   $x_k$  - это равномерное разбиение единичного отрезка от 0 до 1 с шагом  $1/n$ , из которого удалена правая крайняя точка, относительно скалярного произведения.

Доказательство:

$$\begin{aligned} (y^{(j_1)}, y^{(j_2)}) &= \frac{1}{n} \sum_{k=0}^{n-1} y^{(j_1)}(x_k) \overline{y^{(j_2)}(x_k)} = \begin{cases} 1, j_1 = j_2 \\ 0, j_1 \neq j_2 \end{cases} = \begin{bmatrix} x_k = kh \\ y_j(x) = \exp(2\pi i(j_0 + j)x) \end{bmatrix} = \\ &= \frac{1}{n} \sum_{k=0}^{n-1} \exp(2\pi i(j_0 + j)kh) \overline{\exp(2\pi i(j_0 + j)kh)} = \frac{1}{n} \sum_{k=0}^{n-1} \exp(2\pi i(j_0 + j)kh) \exp(-2\pi i(j_0 + j)kh) = \\ &= \frac{1}{n} \sum_{k=0}^{n-1} \exp(2\pi i k(j_1 + j_0 - j_1 - j_0)) = \begin{bmatrix} \text{по формуле геом. прогр} \\ q = \exp(2\pi i h(j_1 - j_2)) \\ 1 + q + \dots + q^{n-1} = \frac{q^n - 1}{q - 1} \end{bmatrix} = \\ &= \frac{1}{n} \frac{\exp(2\pi i(h-1+1)h(j_1 - j_2) - 1)}{\exp(2\pi i h(j_1 - j_2))} = 0 \end{aligned}$$

Рассмотрим  $j_1 = j_2$

$$\frac{1}{n} \sum_{k=0}^{n-1} \exp(0) = 1$$

Используя разложение Фурье, идею тригонометрической интерполяции и выше доказанную теорему, получим следующие формулы тригонометрической интерполяции (при  $x_k = kh$ , где  $h = 1/n$ ,  $k = \overline{0, n-1}$ ).

$$y(x) = \sum_{j=0}^{n-1} A_j - y^{(j)}(x) = \sum_{j=1}^{n-1} A_j \exp(2\pi i(j + j_0)x)$$

$$A_j = (y * y^{(j)}) = \frac{1}{n} \sum_{k=0}^{n-1} y(x_k) \overline{y^{(j)}(x_k)} = \frac{1}{n} \sum_{k=0}^{n-1} y_k \exp(-2\pi i(j + j_0) \frac{k}{n})$$

$$y(x) = \sum_{j=0}^{n-1} A_j \exp(2\pi i(j + j_0)x)$$

Эти формулы применимы только для узлов  $x_k=k/n$ , т.е. равномерного разбиения интервала  $(0,1)$ . Чтобы перейти от этих узлов к равноотстоящим узлам на произвольной сетке, нужно сделать соответствующую замену. В этих формулах необходимо выбрать  $j_0=-(n-1)/2$ . Это сделано для того, чтобы в формулах возникали  $\exp$ :  $\exp(ik)$  и  $\exp(-ik)$ , а при сложении таких  $\exp$  их мнимые части сокращаются. Если в формуле сделать замену по  $x$  и в качестве  $j_0$  взять  $j_0=-(n-1)/2$ , то формулы преобразуются к виду:

$$\begin{cases} y(x) = \sum_{-\frac{n}{2} < j < \frac{n}{2}} A_j \exp(2\pi i j \frac{x-x_0}{nh}) & (3.1) \\ A_j = \frac{1}{n} \sum_{k=0}^{n-1} y_k \exp(-2\pi i \frac{k_j}{n}) & (3.2) \end{cases}$$

Эти формулы применимы для равноотстоящих узлов интерполяции на произвольной сетке, где  $k = \overline{0, n-1}$

### **п 3. Быстрое преобразование Фурье**

Дискретным преобразованием Фурье называется перевод набора значений функции  $(y_0 \dots y_{n-1})$  в набор коэффициентов  $(A_0 \dots A_{n-1})$ , участвующих в разложении Фурье.

Замечание:

$\Rightarrow$  - прямое преобразование Фурье  $(y_0 \dots y_{n-1}) \Rightarrow (A_0 \dots A_{n-1})$

$\Leftarrow$  - обратное преобразование Фурье  $(A_0 \dots A_{n-1}) \Rightarrow (y_0 \dots y_{n-1})$

Задача тригонометрической интерполяции: по набору  $y_i$  получить набор  $A_i$ .

По формуле 3.1 интерполируем функцию в любой точке.

Если использовать формулу 3.2, то для вычисления набора коэффициентов требуется  $cn^2$  действий (для каждого  $A_i$  потребуется  $C \cdot n^2$  операций).

Для ускорения этой процедуры используют быстрое преобразование Фурье.

## **§ 3. Интерполяция кубическими сплайнами**

### **п 1. Кубические сплайны.**

Кубическим сплайном на сетке  $x_0, x_1, \dots, x_n$  называется функция  $S(x)$ , которая:

1. на каждом интервале  $[x_{i-1}, x_i]$ , где  $1 \leq i \leq n$ , функция  $S(x)$  является кубическим многочленом (обозначим его  $S_i(x)$ ).
2. на всем интервале  $[x_0, x_n]$   $S(x)$  – дважды непрерывно дифференцируемая функция
3. на краях интервала вторая производная обращается в ноль.

$$S''(x_0)=S''(x_n)=0$$

Замечание к свойству 2:

Внутри интервала  $[x_{i-1}, x_i]$   $S(x)$ , будучи кубическим многочленом, непрерывно дифференцируема бесконечно много раз. Необходимо, чтобы в узлах  $x_i$ :

а)  $S_i(x_i)=S_{i+1}(x_i)$  - непрерывность  $S$  в точке  $x_i$ ;

б)  $S'_i(x_i)=S'_{i+1}(x_i)$  - непрерывность  $S'$  в точке  $x_i$ ;

в)  $S''_i(x_i)=S''_{i+1}(x_i)$  - непрерывность  $S''$  в точке  $x_i$ .

## Свойство кубического сплайна

### Теорема 3.8

Пусть на сетке  $x_i$ , где  $i = \overline{0, n}$  задана функция  $f$  со своими значениями  $y_i = f(x_i)$ . Тогда среди всевозможных функций  $y(x)$ , интерполирующих  $f$  на сетке  $x_i$  именно для кубического сплайна достигается минимум интеграла энергии

изгиба  $I(y) = \int_{x_0}^{x_n} (y''(x))^2 dx$  Без доказательства:

### Комментарии к этой теореме:

Из математического анализа известно, что радиус кривизны функции  $y(x)$ :

$R(x) = \left[ \frac{y''(x)}{(1 + (y'(x))^2)^{3/2}} \right]^{-1} = k(x)^{-1}$  ( $k(x)$  – кривизна изгиба). Из физики известно, что

при изгибе линейки необходима энергия  $I = \int_a^b \frac{\gamma(x)k^2(x)}{2} dx$  ( $K$  – коэффициент

упругости), если  $\gamma = \text{const}$ , т.е. линейка однородна и  $y'(x) \ll 1$ , тогда  $dl = \sqrt{1 + (y')^2} dx \approx dx$ ,  $k(x) = y''$

Сплайн (англ. – spline) – линейка.

Раньше интерполяцию производили с помощью гибкой металлической линейки, пропускаемую через точку  $(x_i, y_i)$ . Третье свойство сплайна связано с тем, что на концах интервала  $[x_0, x_n]$  линейка не сгибается, а представляет собой прямую.

Исследуем вопрос о единственности сплайна, интерполирующего  $f$  в точке  $x_i$ . Сплайн задается набором из  $n$ -кубических многочленов  $S_i = a_i x^3 + b_i x^2 + c_i x + d_i$ , итого  $4n$  коэффициентов, которые необходимо найти. Сколько уравнений?

1-ая группа уравнений:

$$S_i(x_i) = y_i, \quad i = \overline{1, n} \quad *$$

2-я группа уравнений:

$$S_i(x_{i-1}) = y_{i-1}, \quad i = \overline{1, n} \quad *$$

Т.к. функция  $S$  на всем интервале  $I$  дважды непрерывно дифференцируема, то в смежных точках  $(x_i \dots x_{n-1})$  должны совпадать значения функции, а так же знаки первой и второй производных слева и справа, для того, чтобы функция, а так же первая и вторая производные в этих точках были непрерывны.

$S_-$  - правый предел функции

$S_+$  - левый предел функции

$$S_-(x_i) = S_+(x_i)$$

$$S_i'(x_i) = S_{i+1}'(x_i)$$

Эти условия являются очевидным следствием условий (\*)

$$\left. \begin{aligned} S_-'(x_i) &= S_+'(x_i) \\ S_i'(x_i) &= S_{i+1}'(x_i) \end{aligned} \right\} (n+1) \text{ условий}$$

$$\left. \begin{aligned} S_-''(x_i) &= S_+''(x_i) \\ S_i''(x_i) &= S_{i+1}''(x_i) \end{aligned} \right\} (n+1) \text{ условий}$$

Всего  $(4n-2)$  условий. Последние два условия берутся из 3-его пункта определения кубического сплайна. Итак, для нахождения кубического сплайна, а именно,  $4n$  его коэффициентов, получим  $4n$  уравнений (все эти уравнения нечетные). Можно вычислить кубический сплайн, решая соответствующее уравнение.

На самом деле при вычислении кубического сплайна используется другой алгоритм.

Обозначим  $S''(x_i) = M_i \quad i = \overline{0, n}$

$M_0 = 0 = M_n$

Для неизвестных  $M_i$  система уравнений получится проще и через  $M_i$  нетрудно вычислить и  $S_i(x)$ . Т.к.  $S(x)$  кусочно кубическая функция, то  $S''(x)$  – кусочно линейная функция. Следовательно, на интервале  $[x_{i-1}, x_i]$   $S''$  многочлен первой степени и непрерывная функция.

Следовательно,  $x \in [x_{i-1}, x_i]$ :

$$S''_i(x) = \frac{x - x_i}{x_{i-1} - x_i} M_{i-1} + \frac{x - x_{i-1}}{x_i - x_{i-1}} M_i = M_{i-1} \frac{x_i - x}{h_i} + M_i \frac{x - x_{i-1}}{h_i}, \quad h_i = x_i - x_{i-1}$$

(уравнение прямой проходящей через две точки).

Чтобы получить  $S_i(x)$  проинтегрируем  $S''_i(x)$  дважды.

$$S'_i(x) = -M_{i-1} \frac{(x_i - x)^2}{2h_i} + M_i \frac{(x - x_{i-1})^2}{2h_i} + c$$

$$S_i(x) = M_{i-1} \frac{(x_i - x)^3}{6h_i} + M_i \frac{(x - x_{i-1})^3}{6h_i} + cx + d$$

константу интегрирования находим из краевых условий:

$$\begin{cases} S(x_{i-1}) = -y_{i-1} \\ S(x_i) = y_i \end{cases}$$

$$S_i(x) = M_{i-1} \frac{(x_i - x)^3}{6h_i} + M_i \frac{(x - x_{i-1})^3}{6h_i} + \left( y_{i-1} - \frac{M_{i-1}h_i^2}{6} \right) \frac{x_i - x}{h_i} + \left( y_i - \frac{M_i h_i^2}{6} \right) \frac{x - x_{i-1}}{h_i}, \quad (3.18)$$

Теперь для расчета кубического сплайна необходимо найти неизвестные моменты  $M_i$ . Их находим, используя условия непрерывности первой производной.

$$S_{i-1}'(x_i) = S_i'(x_i) \quad i = \overline{0, n-1}$$

После всех преобразований эти условия превращаются в СЛАУ размера  $(n-1)(n-1)$

$$CM = d, \quad (3.19)$$

где

$$M = \begin{pmatrix} M_1 \\ M_2 \\ \vdots \\ M_{n-1} \end{pmatrix}$$

$$C = \begin{pmatrix} \frac{h_1 + h_2}{3} & \frac{h_3}{6} & 0 & 0 \\ \frac{h_2}{6} & \frac{h_2 + h_3}{3} & \frac{h_3}{6} & 0 \\ 0 & \frac{h_3}{6} & \frac{h_3 + h_4}{3} & \frac{h_4}{6} \end{pmatrix}$$

т.е. элементы матрицы С вычисляются по формулам:

$$C_{ij} = \begin{cases} \frac{h_i + h_{i+1}}{3} & i = j \\ \frac{h_i + 1}{6} & j = i + 1 \\ \frac{h_i}{6} & j = i - 1 \\ 0 & |i - j| \geq 2 \end{cases}$$

$$d = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_{n-1} \end{pmatrix}, \quad \text{где}$$

$$d_i = \frac{y_{i+1}}{h_{i+1}} - \frac{y_i - y_{i-1}}{h_i}$$

Алгоритм расчета кубического сплайна:

1. Формируем матрицу С, вектор-столбец d правых частей в СЛАУ (3.19).
2. решаем СЛАУ (3.19) и находим неизвестные моменты  $M_i$  (например, методом Гаусса);
3. по формуле (3.18) вычисляем значение кубического сплайна в заданной точке  $x$ .

Систему линейных уравнений 3.19 можно решать обычным методом Гаусса. На самом деле в матрице С очень много нулевых элементов, она – трехдиагональная. Для подобных матриц разработан специальный метод – метод прогонки. Приведем формулу метода прогонки для трехдиагональной матрицы:

$$Ax = \alpha$$



$$\begin{pmatrix} a_1 & c_1 & 0 & 0 & 0 & \dots & \dots & \dots \\ b_2 & a_2 & c_2 & 0 & 0 & 0 & \dots & \dots \\ 0 & b_3 & a_3 & c_3 & 0 & 0 & 0 & \dots \\ 0 & 0 & b_4 & a_4 & c_4 & 0 & 0 & 0\dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & b_n & a_n \end{pmatrix} \alpha = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \dots \\ \alpha_n \end{pmatrix}$$

1. прямой этап метода прогонки:

Вычисляем:

$$\alpha_k = \frac{c_k}{a_k + b_k + c_k}$$

$$\beta_k = \frac{\alpha_k - b_k \beta_{k-1}}{a_k + b_k \alpha_{k-1}}$$

$k = 1, 2, \dots, n$ .

При этом полагают  $\alpha_0 = \beta_0 = b_1$ ,  $C_n = 0$

2. на обратном этапе метод прогонки вычисляют:

$$x_k = \alpha_k x_{k+1} + \beta_k \quad x_{n-1} = 0$$

Метод прогонки = метод Гаусса для трехдиагональных матриц:

Трудоемкость метода прогонки  $6n$  действий на прямом ходе и  $2n$  действий на обратном, итого  $8n$  действий.

### **Теорема 3.9**

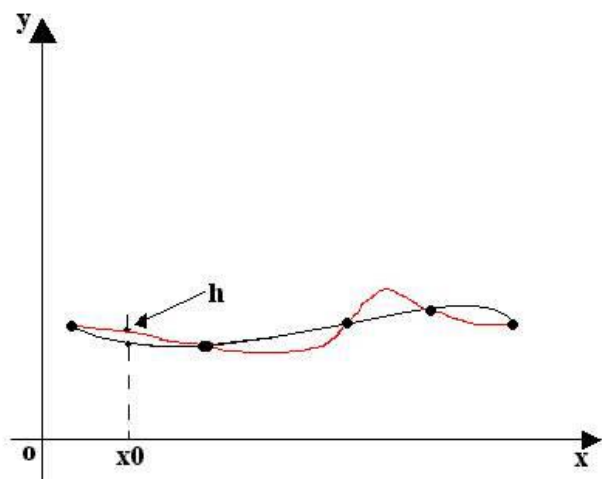
Система линейных уравнений 3.19 всегда имеет единственное решение.

Доказательство:

В матрице  $C$  имеет место диагональное преобладание по строкам. Следовательно, метод простой итерации будет сходиться (для системы 3.19), а как известно он сходится к точному решению. Следовательно, 3.19 всегда имеет решение. Если эта система линейных уравнений  $Ax=d$  с квадратной матрицей  $C$  имеет решение для любого  $d$ , то это решение всегда единственно.

### **п 2. Локальные кубические сплайны.**

Недостаток кубического сплайна – продолжительная затухающая реакция на возмущение.



Замечание:

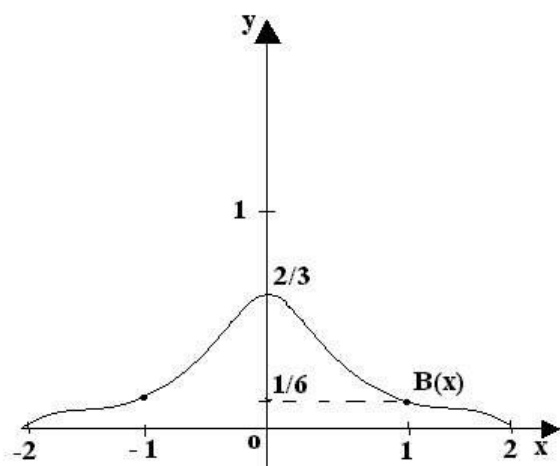
Интерполяционный многочлен реагирует на возмущение заметно хуже. И вообще у кубического сплайна погрешность округления меньше, чем у интерполяционного многочлена. Чтобы избавиться от неограниченно долгой реакции на возмущение вместо обычных кубических сплайнов иногда применяют локальные кубические сплайны.

Стандартным локальным кубическим сплайном называется сплайн

$$B(x) = \begin{cases} \frac{2}{3} - x^2 + \frac{1}{2}x^3 & |x| \leq 1 \\ \frac{1}{6}(2 - |x|)^3 & 1 \leq |x| \leq 2 \\ 0 & |x| \geq 2 \end{cases}$$

Задача:

Доказать, что  $B(x)$  – кубический сплайн – 4 балла.



Лемма:  $B$  – кубический сплайн.

Доказательство:

Необходимо проверить, что для  $B$  выполняются все свойства кубического сплайна (на сетке  $-2, -1, 0, 1, 2$ ).

Очевидно, что на каждом интервале В – кубический многочлен

$$S''(2)=S''(-2)=0$$

В точках  $-2, -1, 0, 1, 2$  функция В дважды непрерывно дифференцируема: для этого необходимо проверить условия в каждой точке

$$S_-(x_i) = S_+(x_i)$$

$$S'_-(x_i) = S'_+(x_i)$$

$$S''_-(x_i) = S''_+(x_i)$$

$$S''_-(1) = S''_+(1)$$

$$S''_+(1) = \left( \frac{1}{6}(2-x)^3 \right)'' \Big|_{x=1} = 1$$

$$S''_-(1) = \left( \frac{2}{3} - x^2 + \frac{1}{2}x^3 \right)' \Big|_{x=1} = 1$$

Пусть  $x_i$  равноотстоящие узлы с шагом  $h$ , тогда локальным кубическим сплайном первого (второго) типов называется функция:

$$B^{(i)}(x) = \sum_{k=-1}^{n+1} \alpha_k^{(i)} B\left(\frac{x-kh}{h}\right), \quad где$$

$\alpha_k^{(i)}$  – вычисляются следующим образом

$i=1$  (локальный кубический сплайн первого типа)

$$\alpha_k^1 = y_k \quad k = \overline{0, n}$$

$$\left. \begin{aligned} \alpha_{-1} &= y_{-1} = 2y_0 - y_1 \\ \alpha_{n+1} &= y_{n+1} = 2y_n - y_{n-1} \end{aligned} \right\} \quad \text{экстраполяция по двум крайним точкам}$$

$i=2$  (локальный кубический сплайн второго типа)

$$\alpha_k^{(2)} = (8y_k - y_{k-1} - y_{k+1})/6 \quad k = \overline{-1, n+1}$$

При этом потребуется по два новых  $y$  слева и справа.

Экстраполяция по четырем точкам

$$\left. \begin{aligned} y_{-1} &= 4y_0 - 6y_1 + 4y_2 - y_3 \\ y_{-2} &= 10y_0 - 20y_1 + 15y_2 - 4y_3 \end{aligned} \right\}$$

$$\left. \begin{aligned} y_{n+1} &= 4y_n - 6y_{n-1} + 4y_{n-2} - y_{n-3} \\ y_{n+2} &= 10y_n - 20y_{n-1} + 15y_{n-2} - 4y_{n-3} \end{aligned} \right\}$$

### **Теорема 3.10**

Значения локального кубического сплайна  $B^{(i)}(x)$  в точках  $x$  зависит только от значений функции в 4 соседних узлах (2 справа и 2 слева) при  $i=1$  и, соответственно, от 6 соседних узлов при  $i=2$  (3 слева и 3 справа)

Доказательство:

Самостоятельно 5 баллов

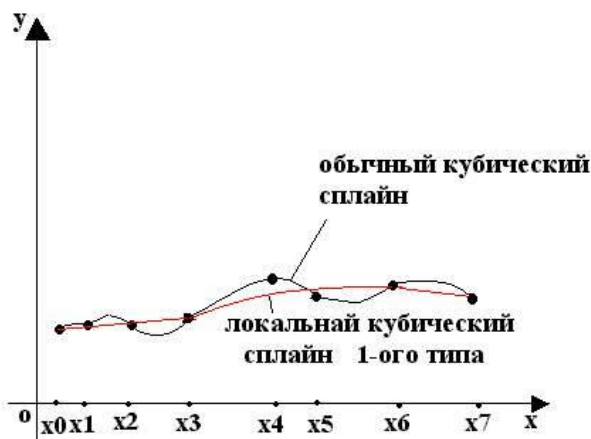
**Вывод:**

Возмущения затухают полностью за 2 шага для локального кубического сплайна 1-ого типа и за 3 шага для локального кубического сплайна 2-ого типа

**Замечание:**

Локальные кубические сплайны обоих типов не интерполируют заданную функцию  $f$ .

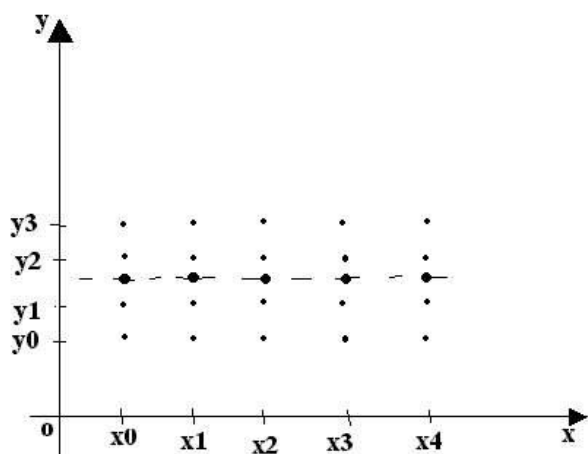
Локальные кубические сплайны аппроксимируют заданную функцию  $f$ , сглаживая ее неровности и скачки. При этом локальные кубические сплайны 1-ого типа лучше сглаживают скачки, но хуже совпадают с функцией  $f$  в узлах интерполяции, а второго типа наоборот, хуже сглаживают скачки, но ближе к функции  $f$  в узлах интерполяции.



#### § 4. Многомерная интерполяция

Рассмотрим функцию 2-ух переменных  $f(x, y)$  заданную на прямоугольной сетке .

Задача: найти значение функции  $f$  в любой заданной точке  $(x, y)$ .



Интерполируя по  $y$  находим значения  $f$  в (\*). После этого интерполируя по  $x$ , используя найденные значения в (\*), находим значение  $f$  в точке, т.е.  $f(x, y)$ .

Замечание:

Предложенный вариант интерполяции зависит от порядка интерполяции (сначала по  $y$ , а потом по  $x$ , как было разобрано или сначала по  $x$ , а потом по  $y$ ). Результат получается разный, но близкий.

Существуют и другие методы многомерной интерполяции.

Метод интерполяции выбирают в зависимости от вида функции и точки  $x$ , в которой необходимо интерполяцию сделать.

Выгодно использовать интерполяцию многочленами, когда функция  $f(x)$  – плавная или узлы расположены достаточно часто, т.е.



Интерполяция кубическим сплайном даст вполне удовлетворительный результат даже для довольно редкой сетки.

Недостаток кубического сплайна – “спрямление” функции около точек  $x_0$  и  $x_n$ .

Недостаток тригонометрической интерполяции: удачно интерполирует только периодические непрерывные функции. Очень часто функцию приводят к такому виду с помощью специальных преобразований.