I Сеточно-разностные методы

1 Векторные и функциональные нормы

Расстояние между точками геометрического пространства обобщается на значительно более сложные *пространства бесконечных последовательностей* и *функциональные пространства* с помощью понятия **нормы**. Норма позволяет определить и вычислить «расстояние» между двумя последовательностями/функциями, которые рассматриваются как две различные «точки» некоторого арифметического/функционального пространства. Для вычислительных методов важны определения норм пространства *конечномерных арифметических векторов*:

Определение I.1: Функционал. Норма вектора. Три основные нормы

Функционал — это числовая функция от векторного аргумента.
Нормой вектора $\vec{u} = (u_0, u_1, \ldots, u_n)^T \in \mathbb{R}^{n+1}$ называется функционал, который удовлетворяет трём *аксиомам нормы*:

- **A1**: $\|\vec{u}\| > 0 \Leftrightarrow \vec{u} \neq \vec{0}, \ \|\vec{0}\| = 0$ — *аксиома положительности*;
- **A2**: $\|\alpha\vec{u}\| = |\alpha| \cdot \|\vec{u}\|, \ \forall\alpha \in \mathbb{R}$ — *аксиома однородности первой степени*;
- **A3**: $\|\vec{u} + \vec{v}\| \leqslant \|\vec{u}\| + \|\vec{v}\|$ — *аксиома неравенства треугольника*.

Основные нормы: $\|\vec{u}\|_\infty \equiv \max\limits_{0 \leqslant i \leqslant n} |u_i|$ — **равномерная норма** (также обозначается $\|\vec{u}\|$),

$\|\vec{u}\|_1 \equiv \sum\limits_{i=0}^{n} |u_i|$ — **норма l_1**, $\quad \|\vec{u}\|_2 \equiv \sqrt{\sum\limits_{i=0}^{n} u_i^2}$ — **норма l_2** или **евклидова норма**.

Определение I.2: Эквивалентные нормы

Две нормы $\|\cdot\|_{\mathrm{I}}$ и $\|\cdot\|_{\mathrm{II}}$ называются эквивалентными, если существуют две положительные константы c_1 и c_2 такие, что для любого вектора $\vec{u}$ выполнено двойное неравенство

$$c_1\|\vec{u}\|_{\mathrm{I}} \leqslant \|\vec{u}\|_{\mathrm{II}} \leqslant c_2\|\vec{u}\|_{\mathrm{I}}$$

Упражнение № I.1

Доказать, что эквивалентность норм является *отношением эквивалентности*. Это означает, что выполнены **три аксиомы эквивалентности**:

- Любая норма эквивалентна самой себе: $\|\vec{u}\|_{\mathrm{I}} \leqslant \|\vec{u}\|_{\mathrm{I}} \leqslant \|\vec{u}\|_{\mathrm{I}}$ — *свойство рефлексивности*;.
- Если норма $\|\cdot\|_{\mathrm{I}}$ эквивалентна норме $\|\cdot\|_{\mathrm{II}}$, то норма $\|\cdot\|_{\mathrm{II}}$ эквивалентна норме $\|\cdot\|_{\mathrm{I}}$ — *свойство симметричности*;
- Если норма $\|\cdot\|_{\mathrm{I}}$ эквивалентна норме $\|\cdot\|_{\mathrm{II}}$, а норма $\|\cdot\|_{\mathrm{II}}$ эквивалентна норме $\|\cdot\|_{\mathrm{III}}$, то норма $\|\cdot\|_{\mathrm{I}}$ эквивалентна норме $\|\cdot\|_{\mathrm{III}}$ — *свойство транзитивности*.

Эквивалентные нормы равноправны в смысле сходимости: если последовательность векторов является сходящейся по одной из норм, то она сходится по всем эквивалентным ей нормам.

Упражнение № I.2

- Показать, что *основные* нормы действительно являются нормами, т. е. удовлетворяют всем трём аксиомам нормы.
- Показать, что нормы $\|\cdot\|_\infty$, $\|\cdot\|_1$ и $\|\cdot\|_2$ являются эквивалентными нормами. Найти константы эквивалентности для этих норм.
- Показать, что если $\|\cdot\|$ — норма, то $\forall K > 0$ функционал $\|\cdot\|' \equiv K\|\cdot\|$ также является нормой.

Определение I.3: Норма функции (норма в функциональном пространстве)

Нормой непрерывной функции (нормой в функциональном пространстве $C[a,b]$) называется функционал $\|\cdot\|$ (функция от аргумента-функции), который определен для всех элементов функционального пространства $u(x), v(x) \in C[a,b]$, и который удовлетворяет трём аксиомам нормы:

- $\|u\| > 0,\ u(x) \not\equiv 0,\ \|u \equiv 0\| = 0$ — *аксиома положительности*;
- $\|\alpha u\| = |\alpha| \cdot \|u\|,\ \forall \alpha \in \mathbb{R}$ — *аксиома однородности первой степени*;
- $\|u + v\| \leqslant \|u\| + \|v\|$ — *аксиома неравенства треугольника*.

Три основные нормы: $\|u\|_\infty \equiv \max\limits_{x\in[a,b]} |u|$ — **равномерная норма** (часто $\|u\|_\infty \equiv \|u\|$),

$$\|u\|_1 \equiv \int\limits_a^b |u| dx \text{ — } \textbf{норма } \boldsymbol{l_1}, \quad \|u\|_2 \equiv \sqrt{\int\limits_a^b u^2(x) dx} \text{ — } \textbf{норма } \boldsymbol{l_2} \text{ или } \textbf{евклидова норма}.$$

Упражнение № I.3

Показать, что *основные* функциональные нормы действительно являются нормами, т. е. удовлетворяют всем трём аксиомам нормы.
Проверить следующие свойства основных норм:

- рассмотреть последовательность функций $\varphi_n(x) = \left\{\boxed{1 - nx}, \text{if } x \in [0; 1/n]; \boxed{0}, \text{if } x \in [1/n; 1]\right\}$ и показать, что нормы $\|\cdot\|_\infty$ и $\|\cdot\|_1$ (а также нормы $\|\cdot\|_\infty$ и $\|\cdot\|_2$) НЕ эквивалентны;
- на отрезке $[0; 1]$ рассмотреть последовательность функций $\varphi_n(x) = \sqrt{x + 1/n}$ и показать, что нормы $\|\cdot\|_1$ и $\|\cdot\|_2$ НЕ эквивалентны;
- $\|u^{(n)}\|_\infty \equiv M_n, \quad \|uv\|_\infty \leqslant \|u\|_\infty \cdot \|v\|_\infty$
- $\|uv\|_1 \leqslant \|u\|_\infty \cdot \|v\|_1 = M_0\|v\|_1, \quad \|uv\|_2 \leqslant \|u\|_\infty \cdot \|v\|_2 = M_0\|v\|_2.$

Определение I.4: Сетка на отрезке. Узлы сетки. Локальный i-ый шаг сетки. Вектор шагов сетки. Шаг сетки. Неравномерные и равномерные сетки

Сеткой размера N на отрезке $[a, b]$ называется пронумерованный набор из $(N+1)$-ной *различных* точек этого отрезка: $\boldsymbol{a = x_0 < x_1 < x_2 < \ldots < x_{N-1} < x_N = b}$. Концы отрезка *всегда* включены в этот набор. Точки сетки также называются **узлами сетки**. Длина отрезка между соседними узлами называется **локальным i-ым шагом сетки** и обозначается $\boldsymbol{h_i = x_i - x_{i-1} > 0}$. Все шаги образуют числовой **вектор шагов сетки $\mathbf{H}$** размерности N. Норма вектора шагов $\|\mathbf{H}\|_\infty \equiv h$ называется **шагом сетки**. Для **равномерной сетки** выполнено: $\boldsymbol{h_1 = h_2 = \ldots = h_N} \equiv \boldsymbol{h = (b-a)/N}$. Иначе сетка является **неравномерной**. Сетка, построенная на $[a, b]$, обозначается как $\mathbf{Grid}[\boldsymbol{a}, \boldsymbol{b}]$ или $\mathbf{Grid_h}[a, b]$. Дополнительный индекс $\boldsymbol{h}$ обычно используют тогда, когда рассматривается множество различных сеток на отрезке.

Таким образом, любые две сетки имеют хотя бы два общих узла (концы отрезка) и пересечение множеств сеточных узлов всегда непусто.

Определение I.5: Операции над сетками: укрупнение, измельчение, сложение

- **Укрупнение сетки** — это операция, при которой часть узлов удаляется, а нумерация оставшихся узлов *сжимается*.
- **Измельчение сетки** — это операция, при которой к сетке добавляются новые узлы, им присваиваются уникальные номера, а старые узлы перенумеруются.
- Операция **сложения двух и более сеток** состоит в объединении двух и более узловых множеств с последующей новой нумерацией узлов.

Определение I.6: Сеточная функция. Сеточная норма

Сеточная функция $\boldsymbol{^h u \equiv \vec{u} \equiv \{u_i\}}$ на сетке $\mathbf{Grid_h}[\boldsymbol{a}, \boldsymbol{b}]$ — это числовой вектор, каждый элемент которого соответствует одному и только одному узлу сетки: $u_i \leftrightarrow x_i, i = 0, 1 \ldots N$. Норма числового вектора $\boldsymbol{\vec{u}}$ называется **сеточной нормой**.

Сеточную функцию также можно определить как **множество пар** вида $\{(x_i, u_i)\}_{i=0\ldots N}$.

Определение I.7: Проекция на сетку функции, которая непрерывна на отрезке

Проекция функции $\boldsymbol{u(x)}$ на сетку $\mathbf{Grid_h}[\boldsymbol{a}, \boldsymbol{b}]$ — это сеточная функция $\boldsymbol{[u] \equiv {^h[u]}}$, составленная из значений функции $u(x)$ в узлах сетки: $\boldsymbol{[u]_i \equiv {^h[u]_i} = u(x_i)}, i = 0 \ldots N$.

Индекс h применяется в обозначениях в тех случаях, когда сеток более одной и рассматриваются *различные* проекции функции на различные сетки.

Важное отличие *сеточной функции* от *проекции функции на сетку* состоит в том, что сеточная функция определена **только** в узлах сетки, а по проекции $[u]$ может быть восстановлена породившая её функция $u(x)$ (в частности, непрерывная функция).

Рассмотрим произвольную непрерывную функцию $u(x)$. В некотором функциональном пространстве U этой функции соответствует значение нормы $\|u(x)\|_U$. Далее рассмотрим бесконечно-измельчаемую систему равномерных сеток $\{\mathrm{Grid}_h[a,b]\}_{h\to 0}$. На каждой сетке определена норма сеточных функций $\|\cdot\|_{\mathrm{Grid}_h}$. Эти нормы можно вычислить от проекций ${}^h[u]$ функции $u(x)$ и получить множество норм $\{\|[u]\|_{\mathrm{Grid}_h}\}$.

Определение I.8: Согласованность функциональной и сеточной норм

Говорят, что множество сеточных норм $\{\|\cdot\|_{\mathrm{Grid}_h}\}$ **согласовано** с функциональной нормой $\|\cdot\|_U$, если для достаточно гладких функций $u(x)$ выполнено:

$$\lim_{h\to 0}\|[u]\|_{\mathrm{Grid}_h} = \|u(x)\|_U$$

Известно, что непрерывная на отрезке функция $u(x)$ полностью определена своими значениями в рациональных точках этого отрезка. Значит, проекций ${}^h[u]$ на все **рациональные сетки** (сетки, состоящие из рациональных узлов) *достаточно* для восстановления *всех* значений функции $u(x)$ на отрезке.

Теперь опять вернёмся к примеру модели «хищник–жертва» (**??**):

$$\begin{cases} \dfrac{dx}{dt} = \alpha x - \beta xy,\ x(0) = x_0, \\ \dfrac{dy}{dt} = -\gamma y + \delta xy,\ y(0) = y_0. \end{cases} \Leftrightarrow \begin{cases} \mathcal{D}\{u\} \equiv \begin{pmatrix} \dot{x} - \alpha x + \beta xy \\ \dot{y} + \gamma y - \delta xy \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix},\ t \in [0,T]. \\ d\{u\} \equiv \begin{pmatrix} x(0) - x_0 \\ y(0) - y_0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \end{cases}$$

здесь $\boldsymbol{\mathcal{D}\{u\}}$ — **дифференциальный оператор**, $\boldsymbol{d\{u\}}$ — **оператор начальных и граничных условий** (который также может быть дифференциальным). Введя равномерную сетку $\mathrm{Grid}_h[0,T]$: $t_i = ih$, $i = 0, 1\ldots, N$, $h = T/N$, мы можем заменить операторы на их **сеточные аналоги**, которые обозначим $\mathcal{D}_h$, и d_h. Таким образом получаем задачу определения числовой матрицы $({}^h u)$ размера $(N+1)\times 2$ из системы алгебраических уравнений: $\boldsymbol{\mathcal{D}_h\{{}^h u\} = (0), d_h\{{}^h u\} = (0)}$ (в данном примере (0) — нулевые матрицы того же размера $(N+1)\times 2$). В случае *линейных* дифференциальных задач часто применяют обозначения $\mathcal{L}\{u\} \equiv \mathcal{D}\{u\}$, $\ell\{u\} \equiv d\{u\}$, и отсюда $\mathcal{L}_h \equiv \mathcal{D}_h$, $\ell_h \equiv d_h$.

Определение I.9: Разностная схема. Разностное решение

Разностная схема — это семейство сеточных задач $\mathcal{D}_h\{{}^h u\} = (0)$, $d_h\{{}^h u\} = (0)$, которые зависят от параметра h. Решение $\boldsymbol{{}^h u = \{u_i\}}$ разностной схемы называется **разностным решением**. Это решение принимается в качестве *приближенного* решения задачи $\mathcal{D}\{u\} = 0$, $d\{u\} = 0$ (здесь 0 — нулевой вектор).

Далее будем изучать в основном равномерные сетки. Однако мы также увидим, что для некоторых задача существует оптимальный набор узлов неравномерной сетки, позволяющий получить требуемое решение наиболее эффективным образом.

2 Разностные производные на сетках

Определение I.10: Разностная производная. Шаблон, валентности шаблона, ранг шаблона. Канонические первые разностные производные

Разностная производная — это определенная для любых сеточных функций $\{u_i\}$ сеточная функция вида $u_i^{(p,q)} = \sum\limits_{j=-q}^{p} C_j(\vec{h}) u_{i+j}$, $q \leqslant i \leqslant N - p$. Коэффициенты $C_j(\vec{h})$ являются функциями *только вектора* $\vec{h}$. Возможные обозначения $\boldsymbol{\vec{u}^{(p,q)}} \equiv \boldsymbol{{}^{h}\vec{u}^{(p,q)}}$.
Шаблоном разностной производной называется набор $p+q+1$ узловых номеров сетки: $(i-q,\ i-q+1\ldots, i+p)$. Параметры $\boldsymbol{p}$ и $\boldsymbol{q}$ называются **валентности шаблона**. Их сумма $\boldsymbol{p+q}$ называется **рангом шаблона** или **рангом разностной производной**.
Канонические первые разностные производные определяются тремя формулами:
$\partial_- u_i \equiv \dfrac{u_i - u_{i-1}}{h_i} \overset{\mathbf{h_i \equiv h}}{=} \dfrac{u_i - u_{i-1}}{h}$ — **первая левая производная**;
$\partial_+ u_i \equiv \dfrac{u_{i+1} - u_i}{h_{i+1}} \overset{\mathbf{h_i \equiv h}}{=} \dfrac{u_{i+1} - u_i}{h}$ — **первая правая производная**;
$\partial_0 u_i \equiv \dfrac{u_{i+1} - u_{i-1}}{h_i + h_{i+1}} \overset{\mathbf{h_i \equiv h}}{=} \dfrac{u_{i+1} - u_{i-1}}{2h}$ — **первая центральная производная**.

С помощью этих трёх канонических первых производных получают практически важные формулы разностных производных более высоких рангов. Например:
$\partial^2_{\pm} u_i \equiv \partial_+(\partial_- u_i) = \dfrac{1}{h_{i+1}}\left(\dfrac{u_{i+1} - u_i}{h_{i+1}} - \dfrac{u_i - u_{i-1}}{h_i}\right) \overset{\mathbf{h_i \equiv h}}{=} \dfrac{u_{i+1} - 2u_i + u_{i-1}}{h^2}$ — **каноническая вторая разностная производная**.

Упражнение № I.4

Проверить свойства канонических разностных производных на сетке $h_i \equiv h$:

- Для $\partial_- u_i$: $p = 0$, $q = 1$, $C_{-1} = -1/h$, $C_0 = 1/h$;
  Для $\partial_+ u_i$: $p = 1$, $q = 0$, $C_0 = -1/h$, $C_1 = 1/h$;
  Для $\partial_0 u_i$: $p = 1$, $q = 1$, $C_{-1} = -1/(2h)$, $C_0 = 0$, $C_1 = 1/(2h)$.
- $\partial_0 u_i = \frac{1}{2}(\partial_- u_i + \partial_+ u_i)$.
- $\partial_+(\partial_- u_i) = \partial_-(\partial_+ u_i) = \partial^2_{\pm} u_i$. Также обозначают: $\partial^2_{\pm} u_i \equiv \partial_+\partial_- u_i$.

Определение I.11: Порядок аппроксимации

Если для всех функций класса $u(x) \in C^m[a,b]$ и для всех равномерных сеток $\mathrm{Grid}_h[a,b]$ выполнено неравенство $\left|{}^{h}[u]_i^{(p,q)} - {}^{h}\left[u^{(S)}\right]_i\right| \leqslant K_u h^r$, то говорят, что $\vec{u}^{(p,q)}$ является **$\boldsymbol{S}$-ой разностной производной с порядком аппроксимации $\boldsymbol{r}$**. Константа K_u зависит от свойств функции $u(x)$, но **не зависит от шага сетки $\boldsymbol{h}$**. Например, в дальнейшем мы будем обосновывать, что $K_u \sim \|u^{(m)}\|_\infty$ или $K_u \sim \|u^{(m+1)}\|_\infty$, где $\boldsymbol{m \equiv S + r}$.

Утверждение I.1

Левая и правая первые разностные производные имеют первый порядок аппроксимации. Центральная первая разностная производная имеет второй порядок аппроксимации.

Для доказательства используем разложение функции в ряд Тейлора в окрестности i-го узла. При этом выполнено: $x_{i-1} = x_i - h$, $x_{i+1} = x_i + h$, $[u]_i \equiv u(x_i)$, $[u']_i \equiv u'(x_i)$.

$$[u]_{i-1} = [u]_i - h[u']_i + \frac{h^2}{2}u''(\theta_-) \Rightarrow \left|\frac{[u]_i - [u]_{i-1}}{h} - [u']_i\right| = \frac{h}{2}\left|u''(\theta_-)\right| \Rightarrow$$

$$\left|\partial_-[u]_i - [u']_i\right| \equiv \left|\frac{[u]_i - [u]_{i-1}}{h} - [u']_i\right| \leqslant \frac{M_2}{2}h,\ M_2 = \|u''(x)\|_\infty \equiv \max_{x\in[a,b]}\left|u''(x)\right|.$$

$$[u]_{i+1} = [u]_i + h[u']_i + \frac{h^2}{2}u''(\theta_+) \Rightarrow \left|\frac{[u]_{i+1} - [u]_i}{h} - [u']_i\right| = \frac{h}{2}\left|u''(\theta_+)\right| \Rightarrow$$

$$\left|\partial_+[u]_i - [u']_i\right| \equiv \left|\frac{[u]_{i+1} - [u]_i}{h} - [u']_i\right| \leqslant \frac{M_2}{2}h\,,\ \boxed{S = 1,\ r = 1,\ K_u = \frac{M_2}{2}}. \qquad \text{(I.1)}$$

$$\begin{cases} [u]_{i-1} = [u]_i - h[u']_i + \dfrac{h^2}{2}[u'']_i - \dfrac{h^3}{6}u'''(\theta_{-1}) \quad \| \times\ C_{-1} = -1/(2h) \\ [u]_{i+1} = [u]_i + h[u']_i + \dfrac{h^2}{2}[u'']_i + \dfrac{h^3}{6}u'''(\theta_1) \quad \| \times C_1 = 1/(2h) \end{cases} \Rightarrow$$

$$\left|\partial_0[u]_i - [u']_i\right| \equiv \left|\frac{[u]_{i+1} - [u]_{i-1}}{2h} - [u']_i\right| = \frac{h^2}{12}\left|u'''(\theta_{-1}) + u'''(\theta_1)\right| \leqslant \frac{M_3}{6}h^2,$$

$$M_3 = \max_{x\in[a,b]}\left|u'''(x)\right| \equiv \|u'''(x)\|_\infty,\ \boxed{S = 1,\ r = 2,\ K_u = \frac{M_3}{6}}.$$

Также представляет интерес ответ на следующий вопрос: для функций какого класса разностная производная *точно* совпадает с проекцией обыкновенной производной? Ответ следует искать среди многочленов. А именно, таких многочленов, которые тождественно обнуляют погрешность аппроксимации. Очевидно следующее утверждение:

Утверждение I.2: О точности аппроксимации порядка r

Разностная производная ${}^h[u]^{(p,q)}$ на любой сетке $\mathrm{Grid}_h[a,b]$ точно совпадает с проекцией производной ${}^h[u^{(S)}]$, если $u(x)$ — многочлен степени не выше $m-1$.

Для доказательства достаточно заметить, что для таких многочленов выполнено $|u^{(m)}(x)| \equiv 0 \Rightarrow \|u^{(m)}\|_\infty = 0$, а значит $K_u = 0$ и $\left|{}^h[u]_i^{(p,q)} - {}^h\big[u^{(S)}\big]_i\right| = 0$.

Рассмотрим примеры решения некоторых задач.

Упражнение № I.5

Определить шаблон, порядок аппроксимации и константу K для третьей разностной производной $\partial_+\partial_+\partial_+ u_i \equiv \partial^3_+ u_i$. Найти класс функций, на которых формула совпадает с проекцией третьей обыкновенной производной.

Здесь решим аналогичную задачу для третьей разностной производной $\partial_+\partial_-\partial_+ u_i$. Сначала определим шаблон производной и коэффициенты $C_j(h)$:

$$\partial_-\partial_+ u_i = \frac{u_{i-1} - 2u_i + u_{i+1}}{h^2} \Rightarrow \partial_+\left(\partial_-\partial_+ u_i\right) = \frac{\partial_-\partial_+ u_{i+1} - \partial_-\partial_+ u_i}{h} =$$

$$= \frac{\frac{u_i - 2u_{i+1} + u_{i+2}}{h^2} - \frac{u_{i-1} - 2u_i + u_{i+1}}{h^2}}{h} = \frac{(u_i - 2u_{i+1} + u_{i+2}) - (u_{i-1} - 2u_i + u_{i+1})}{h^3} =$$

$$= \frac{-u_{i-1} + 3u_i - 3u_{i+1} + u_{i+2}}{h^3} \Rightarrow \boxed{(i-1,\, i,\, i+1,\, i+2)}, \quad p = 2,\, q = 1$$

$$C_{-1} = -1/h^3,\ C_0 = 3/h^3,\ C_1 = -3/h^3,\ C_2 = 1/h^3.$$

$$\begin{cases} [u]_{i-1} = [u]_i - h[u']_i + \dfrac{h^2}{2}[u'']_i - \dfrac{h^3}{6}[u''']_i + \dfrac{h^4}{24}u^{IV}(\theta_{-1}) \quad || \times C_{-1} \\ [u]_i = [u]_i \quad || \times C_0 \\ [u]_{i+1} = [u]_i + h[u']_i + \dfrac{h^2}{2}[u'']_i + \dfrac{h^3}{6}[u''']_i + \dfrac{h^4}{24}u^{IV}(\theta_1) \quad || \times C_1 \\ [u]_{i+2} = [u]_i + 2h[u']_i + 2h^2[u'']_i + \dfrac{4h^3}{3}[u''']_i + \dfrac{2h^4}{3}u^{IV}(\theta_2) \quad || \times C_2 \end{cases}$$

Обозначим $M_4 = \max\limits_{x\in[a,b]} \left|u^{IV}(x)\right| \equiv \|u\|_\infty$ и сложим четыре уравнения:

$$\frac{1}{h^3}\Big(-[u]_{i-1} + 3[u]_i - 3[u]_{i+1} + [u]_{i+2}\Big) = \frac{1}{h^3}\big(-1+3-3+1\big)[u]_i + \frac{1}{h^2}\big(1-3+2\big)[u']_i +$$

$$+\frac{1}{h}\Big(-\frac{1}{2}-\frac{3}{2}+2\Big)[u'']_i + \Big(\frac{1}{6} - \frac{1}{2} + \frac{4}{3}\Big)[u''']_i - h\Big(\frac{1}{24}u^{IV}(\theta_{-1}) - \frac{1}{8}u^{IV}(\theta_1) +$$

$$+\frac{2}{3}u^{IV}(\theta_2)\Big) = [u''']_i + h\Big(-\frac{1}{24}u^{IV}(\theta_{-1}) + \frac{1}{8}u^{IV}(\theta_1) - \frac{2}{3}u^{IV}(\theta_2)\Big).$$

$$\left|\frac{-[u]_{i-1} + 3[u]_i - 3[u]_{i+1} + [u]_{i+2}}{h^3} - [u''']_i\right| \leqslant h\Big(\frac{1}{24} + \frac{1}{8} + \frac{2}{3}\Big)M_4 = \frac{5M_4}{6}h,$$

$$\boxed{S = 3,\ r = 1,\ K_u = \frac{5M_4}{6}},$$ Формула точна на многочленах степени не выше третьей.

Теперь можем сделать два вывода. Во-первых, формула $\partial_+\partial_-\partial_+ u_i$ является **третьей** разностной производной **первого порядка** аппроксимации. И во-вторых, эта формула точна для всех многочленов степени не выше третьей (для них $M_4 \equiv 0$).

Второй тип задач — это задачи на конструирование разностных производных. В этих задачах известны шаблон $(i-q, ..., i+p)$, порядок производной S и порядок аппроксимации r, а требуется определить константы C_j и функции, для которых найденные формулы являются точными.

Упражнение № I.6

Определить формулу **третьей** ($S = 3$) разностной производной на симметричном пятиточечном шаблоне ($i-2$, $i-1$, i, $i+1$, $i+2$) со **вторым** ($r = 2$) порядком аппроксимации.

Подсказка Решение следует начинать с определения порядка производной m остаточного члена ряда Тейлора: $\boldsymbol{m = S + r = 5}$. Поэтому считаем, что $u(x) \in \in C^V[a, b]$ и все ряды Тейлора записываем вплоть до *пятой* производной.

Рассмотрим пример. Найти на четырехточечном шаблоне ($i-1$, i, $i+1$, $i+2$) формулу первой разностной производной третьего порядка аппроксимации.

В данном случае порядок высшей производной равен 4 = 1 + 3. Записываем формулы Тейлора, умножаем их на неизвестные коэффициенты и складываем:

$$\begin{cases} [u]_{i-1} = [u]_i - h[u']_i + \dfrac{h^2}{2}[u'']_i - \dfrac{h^3}{6}[u''']_i + \dfrac{h^4}{24}u^{IV}(\theta_{-1}) \quad || \times\ C_{-1} \\ [u]_i = [u]_i \quad || \times C_0 \\ [u]_{i+1} = [u]_i + h[u']_i + \dfrac{h^2}{2}[u'']_i + \dfrac{h^3}{6}[u''']_i + \dfrac{h^4}{24}u^{IV}(\theta_1) \quad || \times C_1 \\ [u]_{i+2} = [u]_i + 2h[u']_i + 2h^2[u'']_i + \dfrac{4h^3}{3}[u''']_i + \dfrac{2h^4}{3}u^{IV}(\theta_2) \quad || \times C_2 \end{cases} \qquad (*)$$

Приравниваем сумму коэффициентов перед первой производной единице. Перед другими производными сумма коэффициентов равна нулю.

$$\begin{array}{ll} [u]_i|| & \\ [u']_i|| & \\ [u'']_i|| & \\ [u''']_i|| & \end{array} \left.\begin{array}{r} C_{-1} + C_0 + C_1 + C_2 = 0 \\ -hC_{-1} + hC_1 + 2hC_2 = 1 \\ h^2\frac{1}{2}C_{-1} + h^2\frac{1}{2}C_1 + h^2 2C_2 = 0 \\ -h^3\frac{1}{6}C_{-1} + h^3\frac{1}{6}C_1 + h^3\frac{4}{3}C_2 = 0 \end{array}\right\} \left.\begin{array}{r} \hat{C}_{-1} + \hat{C}_0 + \hat{C}_1 + \hat{C}_2 = 0 \\ -\hat{C}_{-1} + \hat{C}_1 + 2\hat{C}_2 = 1 \\ \hat{C}_{-1} + \hat{C}_1 + 4\hat{C}_2 = 0 \\ -\hat{C}_{-1} + \hat{C}_1 + 8\hat{C}_2 = 0 \end{array}\right\} C_j = \frac{\hat{C}_j}{h}$$

Решаем систему и получаем: $\begin{cases} \hat{C}_{-1} = -\frac{1}{3};\ \hat{C}_0 = -\frac{1}{2}; \\ \hat{C}_1 = 1;\ \hat{C}_2 = -\frac{1}{6} \end{cases}$, $\begin{cases} C_{-1} = -\frac{1}{3h};\ C_0 = -\frac{1}{2h}; \\ C_1 = \frac{1}{h};\ C_2 = -\frac{1}{6h} \end{cases}$.

При построении формул разностных производных может произойти *повышение порядка аппроксимации*. В этом случае дополнительно окажется равен нулю коэффициент при старшей производной. Проверим для нашего примера: $-\frac{h^4}{72} + \frac{h^4}{24} - \frac{h^4}{9} \neq 0$. Значит, повышения порядка нет.

Подставим найденные решения в (*), сложим уравнения и получим:

$$\frac{1}{6h}\Big(-2[u]_{i-1} - 3[u]_i + 6[u]_{i+1} - [u]_{i+2}\Big) =$$
$$= [u']_i - h^3\left(\frac{1}{72}u^{IV}(\theta_{-1}) - \frac{1}{24}u^{IV}(\theta_1) + \frac{1}{9}u^{IV}(\theta_2)\right) \Rightarrow$$

$$\left|\frac{-2[u]_{i-1} - 3[u]_i + 6[u]_{i+1} - [u]_{i+2}}{6h} - [u']_i\right| \leqslant h^3\left(\frac{1}{72} + \frac{1}{24} + \frac{1}{9}\right)M_4 \leqslant \frac{M_4}{6}h^3.$$

Формула $\frac{1}{6h}(-2u_{i-1} - 3u_i + 6u_{i+1} - u_{i+2})$ является первой разностной производной на четырехточечном шаблоне $(i-1,\ i,\ i+1,\ i+2)$ с третьим порядком аппроксимации. Формула является точной для всех многочленов степени не выше третьей.

3 Лабораторная работа № 3 Машинный ноль и машинное эпсилон в формулах разностных производных

Формулы разностных производных являются **асимптотически точными**. Это означает, что уменьшением шага сетки h можно в принципе добиться любой близости между значениями разностной производной и проекцией обыкновенной производной. Однако реальные вычисления проводятся с ограниченной (машинной) точностью, которая характеризуется параметром ρ: $\tilde{x} = x(1+\rho_x)$, $\sup_x |\rho_x| = \rho$. Ниже приведены физические относительные ошибки $\big|\partial_+[\sin(x)]_0 - \cos(x_0)\big|/\cos(x_0)$, которые вычислены для $x_0 = 1$ и для различных шагов сетки h. Зависимость от h оказалась *нелинейной*, с характерным минимумом, после которого ошибка начинает расти.

h	$1.5\cdot10^{-5}$	$1.5\cdot10^{-6}$	$1.5\cdot10^{-7}$	$1.5\cdot10^{-8}$	$1.5\cdot10^{-9}$	$1.5\cdot10^{-10}$	$1.5\cdot10^{-11}$
ошибка	$1.2\cdot10^{-5}$	$1.2\cdot10^{-6}$	$1.2\cdot10^{-7}$	$\boxed{1.3\cdot10^{-8}}$	$1.1\cdot10^{-7}$	$7.2\cdot10^{-7}$	$2.0\cdot10^{-6}$

Таблица 1.1. Зависимость от шага сетки h относительной ошибки, возникающей при вычислении первой правой разностной производной для проекции функции $\sin(x)$ в точке $x_0{=}1$.

Чтобы изучить наблюдаемый минимум, определим термины **машинное эпсилон** и **машинный ноль**. Т. к. $\tilde{x} = x(1+\rho_x)$, то $\tilde{1} = 1 + \rho_1$, $|\rho_1| < \rho$. Значит, ρ — наименьшее положительное число, которое при суммировании меняет единицу: $1 + \rho > 1$.

Определение I.12: Машинное эпсилон и машинный ноль

Машинное эпсилон $\boldsymbol{\varepsilon}_{\text{маш}}$ — это наименьшее положительное число, которое при прибавлении к единице меняет результат: $\varepsilon_{\text{маш}} = \min\{\varepsilon > 0 \text{ and } 1.+\varepsilon > 1\}$. Очевидно, $\boldsymbol{\varepsilon_{\text{маш}} = \rho}$.
Машинный ноль $\boldsymbol{\vartheta}_{\text{маш}}$ — это наименьшее положительное число, которое при прибавлении к нулю меняет результат: $\vartheta_{\text{маш}} = \min\{\vartheta > 0 \ \text{ and } \ 0.+\vartheta > 0\}$. Машинный ноль также называют **точностью представления нуля в памяти компьютера**.

Значения машинного нуля и машинного эпсилон определяются архитектурой компьютера, компилятором языка программирования, типами данных, которые использует компилятор. Для GCC-компилятора на IBM-совместимом персональном компьютере порядок машинного нуля определен пределами $\sim 10^{-45} - 10^{-5000}$. Для GCC-компилятора на IBM-совместимой машине порядок машинного эпсилон определен пределами $\sim 10^{-7} - 10^{-20}$. Данные результаты можно получить например с помощью следующей программы:

```
// Машинное эпсилон и машинный ноль

#include <fstream>
using namespace std;

int main(){
     setlocale(LC_ALL, "Russian");
     float epsilon_f = 1.0f, mach_zero_f = 1.0f;
     double epsilon_d = 1., mach_zero_d = 1.;
     long double epsilon_l = 1., mach_zero_l = 1.;
     ofstream fout;
     fout.open("mepsilon.out");
     int n = 0;

     while (1.0f + epsilon_f / 2.0f > 1.0f){ // Вычисление машинного эпсилон float
              epsilon_f = epsilon_f / 2.0f;
              n++;
     }
     fout << "Количество вычислений = " << n;
     fout << ", Машинное эпсилон float = " << epsilon_f << endl;
     n = 0;

     while (1. + epsilon_d / 2. > 1.){ // Вычисление машинного эпсилон double
               epsilon_d = epsilon_d / 2.;
               n++;
      }
      fout << "Количество вычислений = " << n;
      fout << ", Машинное эпсилон double = " << epsilon_d << endl;
      n = 0;

      while (1. + epsilon_l / 2. > 1.){ // Вычисление машинного эпсилон long double
                epsilon_l = epsilon_l / 2.;
                n++;
       }
       fout << "Количество вычислений = " << n;
       fout << ", Машинное эпсилон long = " << epsilon_l << endl;
       n = 0;

       while (mach_zero_f / 2.0f > 0.0f){ // Вычисление машинного нуля float
```

```
            mach_zero_f = mach_zero_f / 2.0f;
            n++;
        }
        fout << "Количество вычислений = " << n;
        fout << ", Машинный ноль = " << mach_zero_f << endl;
        n = 0;

        while (mach_zero_d / 2. > 0.){ // Вычисление машинного нуля double
            mach_zero_d = mach_zero_d / 2.;
            n++;
        }
        fout << "Количество вычислений = " << n;
        fout << ", Машинный ноль = " << mach_zero_d << endl;
        n = 0;

        while (mach_zero_l / 2. > 0.){ // Вычисление машинного нуля long
            mach_zero_l = mach_zero_l / 2.;
            n++;
        }
        fout << "Количество вычислений = " << n;
        fout << ", Машинный ноль = " << mach_zero_l << endl;

        fout.close();
        return 0;
}
/* Результаты работы программы:
Количество вычислений = 23, Машинное эпсилон float = 1.19209e-07
Количество вычислений = 52, Машинное эпсилон double = 2.22045e-16
Количество вычислений = 63, Машинное эпсилон long = 1.0842e-19
Количество вычислений = 149, Машинный ноль float = 1.4013e-45
Количество вычислений = 1074, Машинный ноль double = 4.94066e-324
Количество вычислений = 16445, Машинный ноль long = 3.6452e-4951 */
```

Листинг I.1. Машинные эпсилон и машинный ноль для GCC-компилятора. Использованы данные типов float, double и long double.

Рассмотрим произвольное $A > 0$ и из определения (I.12) получим: $1. + \varepsilon_{\text{маш}} > 1. \Rightarrow A + A\varepsilon_{\text{маш}} > A$. Значит $\boldsymbol{A\varepsilon_{\text{маш}}}$ является **наименьшим положительным числом, которое при суммировании с числом A изменяет его значение**. Далее

$$\forall A > 0: \;\; A(1+\rho_A) = \tilde{A} \equiv A + (\tilde{A} - A) = A\left(1 + \frac{\tilde{A} - A}{A}\right) \Rightarrow \left|\frac{\tilde{A} - A}{A}\right| = |\rho_A| < \rho = \varepsilon_{\text{маш}}.$$

Теперь преобразуем формулу первой правой разностной производной, считая, что проекция производной функции вычислена точно $\left(f\ell([u']_i) = [u']_i\right)$, а проекция функции вычислена приближенно $\left(f\ell([u]_i) = [\tilde{u}]_i\right)$. В преобразованиях также используем ранее выведенное неравенство $\left|\frac{[u]_{i+1} - [u]_i}{h} - [u']_i\right| \leqslant \frac{M_2}{2}h$:

$$\left|\partial_+[\tilde{u}]_i - [u']_i\right| \equiv \left|\frac{[\tilde{u}]_{i+1} - [\tilde{u}]_i}{h} - [u']_i\right| = \left|\frac{[\tilde{u}]_{i+1} - [u]_{i+1}}{h} + \frac{[u]_i - [\tilde{u}]_i}{h} + \right.$$

$$\left. + \frac{[u]_{i+1} - [u]_i}{h} - [u']_i\right| \leqslant \left|\frac{[\tilde{u}]_{i+1} - [u]_{i+1}}{h}\right| + \left|\frac{[u]_i - [\tilde{u}]_i}{h}\right| + \left|\frac{[u]_{i+1} - [u]_i}{h} - [u']_i\right| =$$

$$= \frac{\left|[u]_{i+1}\right|}{h} \cdot \left|\frac{[\tilde{u}]_{i+1} - [u]_{i+1}}{[u]_{i+1}}\right| + \frac{\left|[u]_i\right|}{h} \cdot \left|\frac{[\tilde{u}]_i - [u]_i}{[u]_i}\right| + \left|\frac{[u]_{i+1} - [u]_i}{h} - [u']_i\right| \leqslant$$

$$\leqslant \frac{M_0}{h}\varepsilon_{\text{маш}} + \frac{M_0}{h}\varepsilon_{\text{маш}} + \frac{M_2}{2}h = \frac{2M_0}{h}\varepsilon_{\text{маш}} + \frac{M_2}{2}h \equiv \varepsilon(h).$$

Точка минимума функции $\varepsilon(h)$ находится из уравнения $\dfrac{2M_0}{h_{\min}}\varepsilon_{\text{маш}} = \dfrac{M_2}{2}h_{\min}$ (*Доказать!*), что дает $h_{\min} = 2\sqrt{M_0/M_2\,\varepsilon_{\text{маш}}} \Rightarrow \boxed{h_{\min} \sim \sqrt{\varepsilon_{\text{маш}}}}$. Именно это соотношение проверено в таблице 1.1, которая получена с помощью следующей программы:

```
// Вычисление физической ошибки |∂+[ũ]0 − [u′]0|/|[u′]0| для разных значений h.
/* Результат работы программы:
h = 1.49012e-05, ошибка = 1.16036e-05
h = 1.49012e-06, ошибка = 1.16035e-06
h = 1.49012e-07, ошибка = 1.16202e-07
h = 1.49012e-08, ошибка = 1.278e-08
h = 1.49012e-09, ошибка = 1.09308e-07
h = 1.49012e-10, ошибка = 7.18072e-07
h = 1.49012e-11, ошибка = 2.03986e-06 */

#include <fstream>
#include <math.h>
using namespace std;
double epsilon_d(); // Вычисление машинного эпсилон double
double u(double); // Функция для дифференцирования
double du(double); // Производная функция

int main(){
    setlocale(LC_ALL, "Russian");
    double x0 = 1., u0 = u(x0), du0 = du(x0), h, duh, meps;
    int k = 3, i;
    ofstream fout;
    fout.open("diff.out");
    meps = epsilon_d();
    h = pow(10., double(k))*pow(meps,0.5); // h = 10^k √ε_маш
    for (i = k; i >= -k; h / = 10., i--){
        duh = (u(x0 + h) - u0) / h; // ∂+[ũ]0 = (u(x0 + h) − u(x0))/h
        fout << "h = " << h << ", ошибка = " << fabs((duh - du0) / du0)<<endl;
    }
    fout.close();
```

```
    return 0;}
double epsilon_d(){ // Вычисление машинного эпсилон double
        double eps = 1.;
        while (1. + eps / 2. > 1.) eps = eps / 2.;
        return eps;}
double u(double x){ // функция для дифференцирования
        return sin(x);}
double du(double x){ // Производная функция
        return cos(x);}
```

Листинг I.2. Проверка формулы первой правой разностной производной в окрестности оптимального шага сетки.

Упражнение № I.7

Изменить программный код так, чтобы он определял *вычислительную* относительную ошибку $\left|\partial_+[\tilde{u}]_0 - [u']_0\right| / \left|\partial_+[\tilde{u}]_0\right|$. Убедиться, что результаты двух программ практически совпадают (для функции $u(x) = \sin(x)$ и для точки $x_0 = 1$).

3.1 Порядок выполнения лабораторной работы № 3 и варианты заданий

- Для заданных значений: S — порядок обыкновенной производной, (p, q) — шаблон разностной производной, r — порядок аппроксимации, получить обоснованную формулу разностной производной. Вычислить константу K_u.
- Для найденной формулы получить теоретическую оценку $h_{\min} \sim (\varepsilon_{\text{маш}})^{1/k}$.
- На языке **Си** написать программу вычисления $\varepsilon_{\text{маш}}$. Далее, для трёх функций проверить полученную ранее теоретическую оценку $h_{\min}$ в точке $x_0 = 1.59$.

Вариант №	Ранг S	Шаблон (p, q)	Порядок r	Теоретический $h_{\min} \sim (\varepsilon_{\text{маш}})^{\frac{1}{k}}$	Расчетное значение $h_{\min}$		
					$\sin(x)$	$10^5 \sin(x)$	$\mathrm{tg}(x)$
1	1	(1,1)	2				
2	2	(1,1)	2				
3	1	(2,0)	2				
4	2	(2,0)	1				
5	1	(0,2)	2				
6	2	(0,2)	1				
7	1	(2,1)	3				
8	3	(2,1)	1				
9	1	(1,2)	3				
10	3	(1,2)	1				