ЯРОШЕВИЧ В.А.

ЧИСЛЕННЫЕ МЕТОДЫ

Лекции, практические занятия, лабораторный практикум

Версия 1.26 (17.04.2017)

МПиТК, 2-ой курс

МИЭТ — 2017

Оглавление

Глава 1

Лекции

1.1 Лекция 1

Редкая техническая, инженерная, физическая задача может быть решена точно и иметь «красивый» ответ. В практической деятельности человеку часто бывает *достаточно* иметь приближение к точному ответу с некоторой допустимой погрешностью.

Поэтому, разработка способов вычислений волновало умы многих видных математиков. Можно упомянуть методы, названные в честь Ньютона, Лагранжа, Чебышёва.

Итак, численные методы имеют дело с приближёнными вычислениями. Важно понимать какая ошибка может быть в итоге допущена. Основные источники ошибок:

1. Ошибка математической модели. Например, планету Земля можно рассматривать как плоскость, если речь идет о расстояниях < 25 км.

В других случаях следует рассматривать Землю как шар со радиусом 6371 км. Более точная модель будет эллипсоидом (большая полуось 6378 км, малая полуось 6357 км). Самая точная модель — это геоид (напоминает форму груши). Все модели содержат погрешность, которой можно пренебречь в одном случае и нельзя в другом.

2. Погрешность измерений. Например, неточность приборов, плохие измерения, неудачная статистическая выборка.

3. Погрешность численного метода. Например, интеграл заменяется конечной суммой ($\int \to \sum$), производная заменяется разделённой разностью ($f' \to \Delta y/\Delta x$).

4. Ошибка округления. Как правило целочисленных вычислений не хватает для практических нужд. Приходится работать с действительными числами $\mathbb{R}$. Компьютер воспринимает числа в позиционной системе с основанием 2. Понятно, что в этом случае иррациональные числа и часть рациональных чисел невозможно точно представить двоичным числом *конечной* длины. Кроме того, если под число отводится фиксированное число ячеек в памяти компьютера, то некоторые близкие числа $x, y \in \mathbb{R}$ станут неотличимы в памяти компьютера (действительно, в 8 байтах можно разместить не более 2^{64} различных чисел, а $|\mathbb{R}| = \infty$). В итоге, уже при вводе данных в компьютер будет допущена ошибка округления (в MATLAB около 15–16 десятичных разрядов). Задача математика состоит в том, чтобы по окончании вычислений финальная относительная ошибка в ответе имела порядок малости близкий к порядку малости относительной ошибки исходных данных, либо не превышала заданный порог ε.

Рассмотрим более подробно последний пункт.

Число в позиционной системе счисления с основанием p представляет из себя слово в алфавите $\{a_0, a_1, \ldots, a_{p-1}, \text{запятая}\}$, где a_i принято называть цифрами. Запятая может встречаться только один раз. Она отделяет целую часть числа от дробной. Чем больше позиций между цифрой и за-

пятой, тем больший (если цифра слева от запятой) или меньший (если цифра справа от запятой) «вес» имеет цифра в числе. То есть важны не только цифры, но и их позиции. Отсюда происходит название «позиционная».

Примеры:

$$123,45_{10} = \underbrace{1\cdot 10^2 + 2\cdot 10^1 + 3\cdot 10^0}_{\text{целая часть}} + \underbrace{4\cdot 10^{-1} + 5\cdot 10^{-2}}_{\text{дробная часть}},$$
$$1F_{16} = 1\cdot 16^1 + 15\cdot 16^0,$$
$$101,11_2 = \underbrace{1\cdot 2^2 + 0\cdot 2^1 + 1\cdot 2^0}_{\text{целая часть}} + \underbrace{1\cdot 2^{-1} + 1\cdot 2^{-2}}_{\text{дробная часть}}.$$

Отклонение по абсолютной величине точного значения некоторой величины x от приближённого значения $\tilde{x}$ называется *абсолютной ошибкой* $\Delta x = |x - \tilde{x}|$. Отношение $|x - \tilde{x}|/|x|$ абсолютной ошибки к величине x называется *относительной погрешностью*.

Округление при сложении. Пусть требуется найти сумму пяти четырехразрядных чисел: $S = 0.2764 + 0.3944 + 1.475 + 26.46 + 1364$. Складывая все эти числа, а затем округляя полученный результат до *четырёх* значащих цифр, получаем $S = 1393$. Однако при вычислении на компьютере округление происходит после каждого сложения. Предполагая условно мантиссу четырехразрядной, проследим за вычислением на компьютере суммы чисел от наименьшего к наибольшему, т. е. в порядке их записи: $0.2764 + 0.3944 = 0.6708$, $0.6708 + 1.475 = 2.156$, $2.156 + 26.46 = 28.62$, $28.62 + 1364 = 1393$; получили $S_1 = 1393$, т. е. верный результат. Изменим теперь порядок вычислений и начнём складывать числа последовательно от последнего к первому: $1364 + 26.46 = 1390$, $1390 + 1.475 = 1391$, $1391 + 0.3944 = 1391$, $1391 + 0.2764 = 1391$; здесь окончательный результат $S_2 = 1391$, он менее точный. Вывод: *в сумме слагаемые желательно упорядочить в порядке возрастания их абсолютных величин.*

Рассмотрим важный пример — использование рядов для вычисления значений функций. Запишем, например, разложение функции $\sin x$ по сте-

пеням аргумента:

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$$

За приближённое значение $\sin x$ можно принять сумму первых N слагаемых ряда. При этом остаток суммы ряда не должен превышать величину погрешности. Это означает, что слагаемое

$$u_k = \frac{x^{2k-1}}{(2k-1)!} \tag{1.1}$$

для $k > N$ тоже должно быть мало. Попробуем вычислить одно и то же значение синуса для различных аргументов:

$$\sin\frac{\pi}{6} = \sin\left(\frac{\pi}{6} + 2\pi\right) = \sin\left(\frac{\pi}{6} + 8\pi\right) = \frac{1}{2}.$$

Заметим, что $\dfrac{\pi}{6} \approx 0,5235$, $\dfrac{\pi}{6} + 2\pi \approx 6.8068$, $\dfrac{\pi}{6} + 8\pi \approx 25.6563$. Для выполнения условия $u_k \to 0$, необходимо, чтобы факториал в знаменателе превысил числитель. На рис. 1.1 столбиками обозначены слагаемые u_k. Хорошо видно, что чем больше аргумент x, тем длиннее будет начальная сумма ряда для заданной точности. В случае $x = \pi/6$ достаточно уже $u_1 + u_2 + u_3$. Для $\pi/6 + 2\pi$ потребуется $u_1 + \dots + u_{11}$ или даже больше. Для $\pi/6 + 8\pi$ ситуация ещё хуже, нужно более 20 слагаемых. Очевидно, для вычисления $\sin x$ выгодно иметь дело с самым малым возможным аргументом. Кроме этого при больших аргументах слагаемые ряда имеют существенно различный порядок (например, для $x = \frac{\pi}{6} + 8\pi$ это $u_1 \sim 10^0$ и $u_{14} \sim 10^{10}$). Это может вызвать потерю значащих цифр при округлении значения мантиссы.

Для тригонометрических функций можно использовать формулы приведения, благодаря чему аргумент будет находится на отрезке $[0, 1]$. При вычислении экспоненты аргумент x можно разбить на сумму целой и дробной частей ($e^x = e^{n+a} = e^n \cdot e^a$, $0 < a < 1$) и использовать разложение в ряд только для e^a, а e^n вычислять умножением. Таким образом, при организации вычислений можно своевременно обойти «подводные камни», дающие потерю точности.

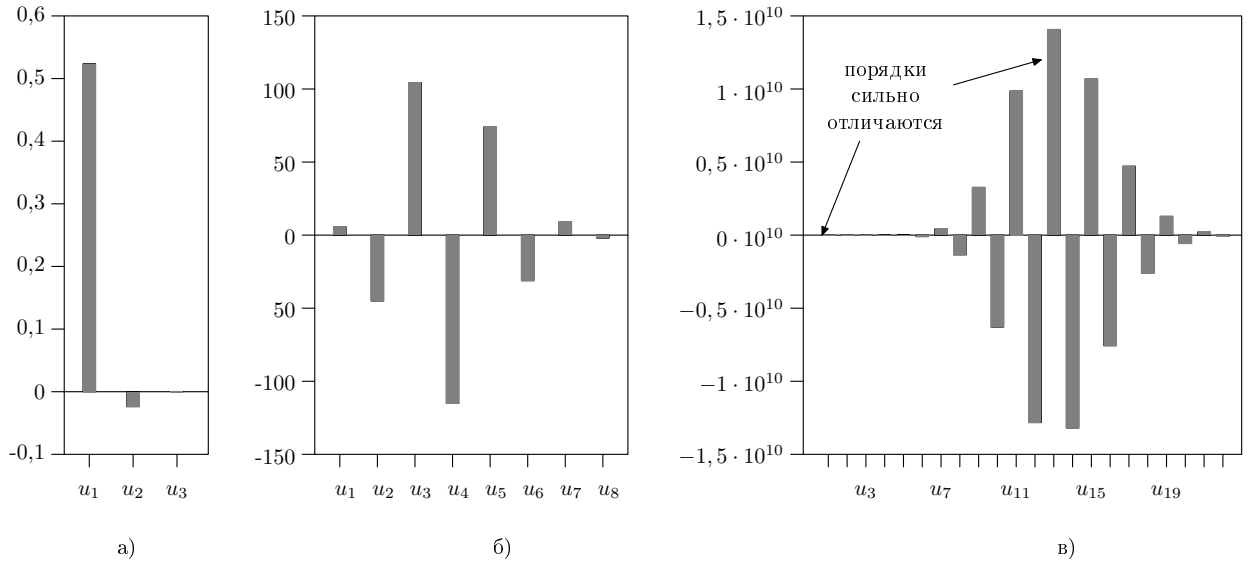


Рис. 1.1: Слагаемые в начальных суммах разложения $\sin x$: а) $x = \frac{\pi}{6} \approx 0,5235$; b) $x = \frac{\pi}{6} + 2\pi \approx 6.8068$; c) $x = \frac{\pi}{6} + 8\pi \approx 25.6563$.

Устойчивость. Пусть в результате решения задачи по исходному значению некоторой величины x находится значение искомой величины y. Если исходная величина имеет абсолютную погрешность Δx, то решение имеет погрешность Δy. Задача называется *устойчивой* по исходному параметру x, если решение y непрерывно от него зависит, т. е. малое приращение исходной величины Δx приводит к малому приращению искомой величины Δy. Другими словами, малые погрешности в исходной величине приводят к малым погрешностям в решении.

Отсутствие устойчивости означает, что даже незначительные погрешности в исходных данных приводят к большим погрешностям в решении или даже к неверному результату. О неустойчивых задачах также говорят, что они чувствительны к погрешностям исходных данных.

Приведем пример неустойчивой задачи. Рассмотрим квадратное уравнение с параметром a

$$x^2 - 2x + \operatorname{sign} a = 0, \operatorname{sign} a = \begin{cases} 1, & a \geqslant 0, \\ -1, & a < 0. \end{cases}$$

Решение этого уравнения в зависимости от значения a таково: $x_1 = x_2 = 1$ при $a \geqslant 0$; $x_{1,2} = 1 \pm \sqrt{2}$ при $a < 0$. Очевидно, что при $a = 0$ сколь угодно

малая отрицательная погрешность в задании a приведет к *конечной*, а не сколь угодно малой погрешности в решении уравнения.

Определение 1. *Задача называется поставленной корректно, если для любых значений исходных данных из некоторого класса её решение 1) существует, 2) единственно и 3) устойчиво по исходным данным.*

Неустойчивость методов. Иногда при решении корректно поставленной задачи может оказаться неустойчивым метод её решения. Такие случаи имели место при вычислении синуса большого аргумента, когда был получен результат, не имеющий смысла. Рассмотрим ещё один пример неустойчивого алгоритма. Построим численный метод вычисления интеграла

$$I_n = \int\limits_0^1 x^n e^{x-1}\,dx, \quad n = 1, 2, \ldots$$

Интегрируя по частям, находим

$$\begin{aligned}
&I_1 = \int\limits_0^1 xe^{x-1}\,dx = xe^{x-1}\Big|_0^1 - \int\limits_0^1 e^{x-1}\,dx = \frac{1}{e},\\
&I_2 = \int\limits_0^1 x^2e^{x-1}\,dx = x^2e^{x-1}\Big|_0^1 - 2\int\limits_0^1 xe^{x-1}\,dx = 1 - 2I_1,\\
&\ldots\\
&I_n = \int\limits_0^1 x^ne^{x-1}\,dx = x^ne^{x-1}\Big|_0^1 - n\int\limits_0^1 x^{n-1}e^{x-1}\,dx = 1 - nI_n.
\end{aligned}$$

Заметим, что подынтегральная функция на всем отрезке интегрирования неотрицательна, следовательно, и значение интеграла — положительное число. Более того, подынтегральная функция на данном интервале ограничена функцией $y = 1$, т.е. значение интеграла не может превышать единицы. Однако если вычислить по этой формуле значение интеграла то результат будет неверным.

На рис. 1.2 n-ый столбик обозначает I_n. Причём чёрный столбик — это I_n, вычисленный по только что изложенному рекурсивному методу, а серый — это I_n, вычисленный по более устойчивому методу конечных сумм.

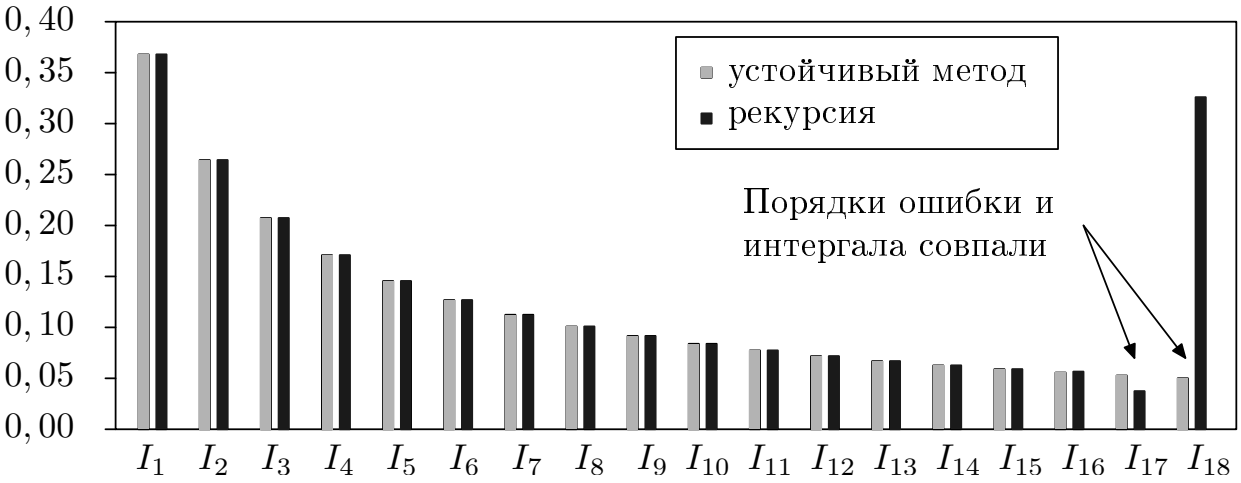
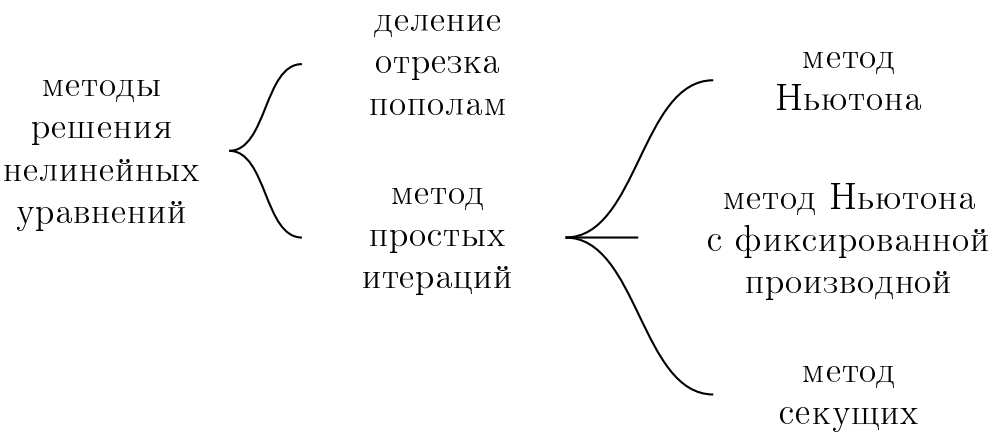


Рис. 1.2: Высичление интеграла устойчивым методом и рекурсией

Видно, что I_{17} немного отклоняется от точного решения, а I_n при $n \geqslant 18$ уже нельзя считать решением. Исследуем источник погрешности. Максимальная абсолютная погрешность при вычислении I_1 равна $0,5 \cdot 2^{53} \approx 5 \cdot 10^{-17}$ (компьютер «обрезал» иррациональное число $1/e$ до 16 десятичных разрядов мантиссы). Однако на каждом этапе эта погрешность умножается на число, модуль которого больше единицы $(-2, -3, \ldots, -18)$, что в итоге даёт $18! \approx 6,4 \cdot 10^{15}$. Это и приводит к результату, не имеющему смысла. Здесь снова причиной накопления погрешностей является алгоритм решения задачи, который оказался неустойчивым.

1.2 Лекция 2



1.2.1 Метод деления отрезка пополам.

Другое название метода — *дихотомия* от греческих слов $\delta\iota\chi\alpha$ «надвое» и $\tau o\mu\eta$ «деление».

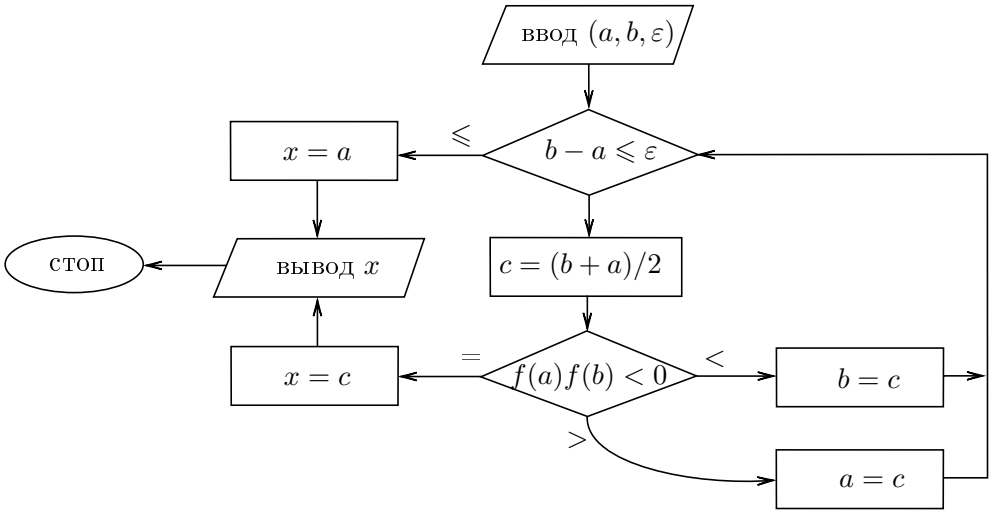


Рис. 2.1: Блок–схема метода деления отрезка пополам

После каждой итерации длина отрезка сокращается вдвое. Следовательно, на n-ой итерации длина отрезка будет $(b - a)/2^n$. Если задана точность $\varepsilon = |x_b - x_*|$, с которой нужно определить корень, то справедливо, что $\varepsilon \leqslant (b - a)/2^n$. Можно оценить число итераций n для достижения точности ε:

$$n = \left[\log_2 \frac{b - a}{\varepsilon}\right] + 1,$$

где квадратные скобки $[\,\cdot\,]$ обозначают целую часть числа (например, $[\pi] = 3$, $[-2{,}123] = -2$).

1.2.2 Метод простых итераций (МПИ).

Заменим уравнение $f(x) = 0$ эквивалентным ему уравнением $x = \varphi(x)$. Это можно сделать многими способами, например, положив $\varphi(x) = x + \psi(x)f(x)$, где $\psi(x)$ — произвольная непрерывная знакопостоянная функция. Выберем некоторое нулевое приближение x_0 и вычислим дальнейшие приближения по формулам

$$x_{n+1} = \varphi(x_n), \qquad n = 0, 1, 2, \ldots$$

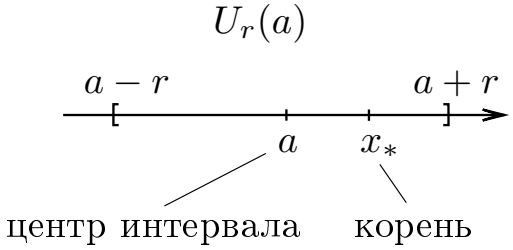
Очевидно, если x_n стремится к некоторому пределу x_*, то этот предел есть корень исходного уравнения.

Условия сходимости.

Определение 2.1. *Функция $s(x)$ называется липшиц-непрерывной с постоянной q на множестве X, если для всех $x', x'' \in X$ выполняется неравенство*

$$|s(x') - s(x'')| \leqslant q|x' - x''|. \tag{2.1}$$

В дальнейшем в качестве X будем брать отрезок $U_r(a) = \{x : \ |x - a| \leqslant r\}$ длины $2r$ с серединой в точке a. Основные свойства МПИ перечислены в следующей теореме.



Теорема 2.2. *Если $\varphi(x)$ липшиц-непрерывна с постоянной $q \in (0, 1)$ на отрезке $U_r(a)$, причем $|\varphi(a) - a| \leqslant (1 - q)r$, то уравнение $x = \varphi(x)$ при любом начальном приближении $x_0 \in U_r(a)$:*

1) имеет на отрезке $U_r(a)$ единственное решение;

2) метод простой итерации $x_{n+1} = \varphi(x_n)$ сходится к x_*;

3) для погрешности справедлива оценка

$$|x_k - x_*| \leqslant q^k |x_0 - x_*|, \qquad k = 0, 1, 2, \ldots \tag{2.2}$$

Доказательство. Сначала докажем по индукции, что $x_k \in U_r(a)$, $k = 1, 2, \ldots$, т. е. что метод простой итерации не выводит за пределы того множества, на котором $\varphi(x)$ липшиц-непрерывна с постоянной $q \in (0, 1)$. Предположим, что $x_j \in U_r(a)$ при некотором $j \geqslant 0$, и докажем, что тогда $x_{j+1} \in U_r(a)$. Из равенства

$$x_{j+1} - a = \varphi(x_j) - a = (\varphi(x_j) - \varphi(a)) + (\varphi(a) - a)$$

получим

$$|x_{j+1} - a| \leqslant |\varphi(x_j) - \varphi(a)| + |\varphi(a) - a|.$$

Учитывая условие липшиц-непрерывности, предположение индукции и условие $|\varphi(a) - a| < (1 - a)r$, имеем

$$|\varphi(x_j) - \varphi(a)| \overset{\text{л.-непр.}}{\leqslant} q|x_j - a| \overset{\text{предп. инд.}}{\leqslant} qr,$$

$$|x_{j+1} - a| \leqslant qr + (1 - q)r \leqslant r,$$

т.е. $x_{j+1} \in U_r(a)$.

Оценим теперь разность двух соседних итераций $x_{j+1} - x_j$. Имеем

$$x_{j+1} - x_j = \varphi(x_j) - \varphi(x_{j-1}),$$

и поскольку все точки x_j, $j = 1, 2, \ldots$, находятся на отрезке $U_r(a)$, получаем оценку

$$|x_{j+1} - x_j| \overset{\text{л.-непр.}}{\leqslant} q|x_j - x_{j-1}| \overset{\text{л.-непр.}}{\leqslant} q^2|x_{j-1} - x_{j-2}| \leqslant \ldots$$

и, следовательно,

$$|x_{j+1} - x_j| \leqslant q^j |x_1 - x_0|, \qquad j = 1, 2, \ldots \tag{2.3}$$

Оценка (2.3) позволяет доказать фундаментальность последовательности $\{x_k\}$. Действительно, пусть p — любое натуральное число. Тогда

$$x_{k+p} - x_k = \sum_{j=1}^{p} (x_{k+j} - x_{k+j-1}),$$

и согласно (2.3) имеем т. е.

$$|x_{k+p} - x_k| \leqslant |x_1 - x_0| \sum_{j=1}^{p} q^{k+j-1} = q^k \frac{1-q^p}{1-q} |x_1 - x_0| \leqslant \frac{q^k}{1-q} |x_1 - x_0|,$$

т.е.

$$x_{k+p} - x_k \leqslant \frac{q^k}{1-q} |x_1 - x_0|, \qquad k, p = 1, 2, \ldots$$

Поскольку правая часть последнего неравенства стремится к нулю при $k \to \infty$ и не зависит от p, последовательность $\{x_k\}$ является фундаментальной. Следовательно, существует

$$\lim_{k\to\infty} x_k = x_* \in U_r(a).$$

Переходя в $x_{n+1} = \varphi(x_n)$ к пределу при $k \to \infty$ и учитывая непрерывность функции $\varphi(x)$, получим $x_* = \varphi(x_*)$, т. е. x_* — решение уравнения $x = \varphi(x)$.

Предположим, что x'_* — какое-то ещё решение уравнения $x = \varphi(x)$, принадлежащее отрезку $U_r(a)$. Тогда

$$x_* - x'_* = \varphi(x_*) - \varphi(x'_*)$$

и по условию теоремы

$$|x_* - x'_*| \leqslant q|x_* - x'_*|.$$

Так как $q < 1$, последнее неравенство может выполняться лишь при $x'_* = x_*$, т. е. решение единственно.

Докажем оценку погрешности (2.2). Итерационное соотношение даёт

$$x_{k+1} - x_* = \varphi(x_k) - \varphi(x_*),$$

и так как $x_k, x_* \in U_r(a)$, приходим к неравенству

$$|x_{k+1} - x_*| \leqslant q|x_k - x_*|,$$

справедливому для всех $k = 0, 1, \ldots$, из которого и следует оценка (2.2). □

Следствие (1). *Если $|\varphi'(x)| \leqslant q < 1$ для всех $x \in U_r(a)$, выполнено условие $|\varphi(a) - a| \leqslant (1-q)r$ и $x_0 \in U_r(a)$, то уравнение $x = \varphi(x)$ имеет единственное решение $x_* \in U_r(a)$, метод $x_{n+1} = \varphi(x_n)$ сходится и справедлива оценка $|x_k - x_*| \leqslant q^k |x_0 - x_*|$, $k = 0, 1, 2, \ldots$*

Доказательство. Воспользуемся формулой конечных приращений:

$$\varphi(x') - \varphi(x'') = (x' - x'') \cdot \varphi(\xi), \qquad \xi \in (x', x'').$$

Следовательно, $\varphi(x)$ является липшиц-непрерывной на $U_r(a)$. Все условия теоремы выполняются. □

Следствие (2)**.** *Пусть уравнение $\varphi(x) = x$ имеет решение x_*, функция $\varphi(x)$ непрерывно дифференцируема на отрезке*

$$U_r(x_*) = \{x : \ |x - x_*| \leqslant r\}$$

и $|\varphi'(x_*)| < 1$. Тогда существует $\varepsilon > 0$ такое, что на отрезке $U_r(x_*)$ уравнение $\varphi(x) = x$ не имеет других решений и метод $x_{n+1} = \varphi(x_n)$ сходится, если только $x_0 \in U_\varepsilon(x_*)$.

Доказательство. Поскольку $\varphi(x)$ непрерывно дифференцируема на отрезке $U_r(x_*)$ и $|\varphi'(x_*)| < 1$, найдутся числа $q \in (0, 1)$ и $\varepsilon \in (0, r]$ такие, что $|\varphi'(x)| \leqslant q < 1$ для всех $x \in U_\varepsilon(x_*)$. □

Оба следствия говорят о сходимости $\{x_n\}$ к корню x_*. При этом следствие 1 гарантирует существование и единственность корня в области $U_r(a)$. Следствие 2, наоборот, требует от нас уверенности, что корень есть в области, где $|\varphi'(x)| < 1$.

Критерий останова. Вблизи корня итерации сходятся примерно как геометрическая прогрессия со знаменателем $q = (x_n - x_{n-1})/(x_{n-1} - x_{n-2})$. Чтобы сумма дальнейших её членов не превосходила ε, должен выполняться критерий сходимости

$$\left| q\frac{x_n - x_{n-1}}{1 - q} \right| = \frac{(x_n - x_{n-1})^2}{|2x_{n-1} - x_n - x_{n-2}|} < \varepsilon.$$

При выполнении этого условия итерации можно прекращать.

1.2.3 Метод Ньютона.

Пусть задано уравнение $f(x) = 0$. Запишем начальную сумму ряда Тейлора для $f(x)$:

$$f(x) \approx f(x_0) + (x - x_0)f'(x_0).$$

Заменим в исходном уравнении функцию $f(x)$ полученным приближением:

$$f(x_0) + (x - x_0)f'(x_0) = 0.$$

Если теперь выразить x и затем сделать замену $x_0 \to x_n$, $x \to x_{n+1}$, получим итерационный процесс

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}. \tag{2.4}$$

Мы получили определение метода *Ньютона* или метода *касательных*. Последнее название обусловлено работой метода:

- из начального приближения x_n строим перпендикуляр к оси абсцисс до пересечения с графиком функции $f(x)$;
- через полученную точку пересечения проводим касательную к графику $f(x)$ до пересечения с осью абсцисс (точка x_{n+1});
- повторяем все сначала ...

Предполагая, что $f(x)$ дважды непрерывно дифференцируема, напишем разложение в ряд Тейлора в корне x_* в окрестности n-го приближения:

$$f(x_*) = 0 = f(x_n) + f'(x_n)(x_* - x_n) + \frac{1}{2}f''(\xi_n)(x_* - x_n)^2,$$

где $\xi_n \in [x_*, x_n]$.

Разделив последнее соотношение на $f'(x_n)$ и перенеся первые два слагаемых из правой части в левую, получим:

$$\left[x_n - \frac{f(x_n)}{f'(x_n)}\right] - x_* = \frac{1}{2}\frac{f''(\xi_n)}{f'(x_n)}(x_n - x_*)^2,$$

что, учитывая (2.4), переписываем в виде

$$x_{n+1} - x_* = \frac{1}{2}\frac{f''(\xi_n)}{f'(x_n)}(x_n - x_*)^2.$$

Отсюда

$$|x_{n+1} - x_*| = \frac{1}{2}\frac{|f''(\xi_n)|}{|f'(x_n)|}|x_n - x_*|^2. \tag{2.5}$$

Получаем оценку

$$|x_{n+1} - x_*| \leqslant \frac{1}{2}\frac{M_2}{m_1}|x_n - x_*|^2,$$

где $M_2 = \max\limits_{[a,b]} |f''(x)|$, $m_1 = \min\limits_{[a,b]} |f'(x)|$.

Очевидно, ошибка на каждом шаге убывает, если

$$\frac{1}{2}\frac{M_2}{m_1}|x_0 - x_*| < 1$$

Требуется хорошее начальное приближение. На рис. 2.2 видно, как из-за плохого начального приближения метод Ньютона зацикливается.

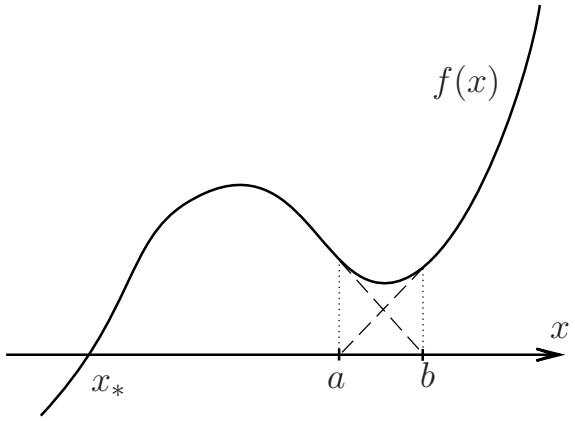


Рис. 2.2: Пример плохого приближения в методе Ньютона

Теорема 2.3. *Пусть задана функция $f(x)$ и определён итерационный процесс $x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$. Если для всех $x \in [a, b]$ справедливо одно из следующих:*

1) $f'(x) > 0$, $f''(x) > 0$ и $x_0 = b$,
2) $f'(x) < 0$, $f''(x) < 0$ и $x_0 = b$,

тогда $\{x_k\}$ монотонно убывает и сходится к x_*;

3) $f'(x) > 0$, $f''(x) < 0$ и $x_0 = a$,
4) $f'(x) < 0$, $f''(x) > 0$ и $x_0 = a$,

тогда $\{x_k\}$ монотонно возрастает и сходится к x_*.

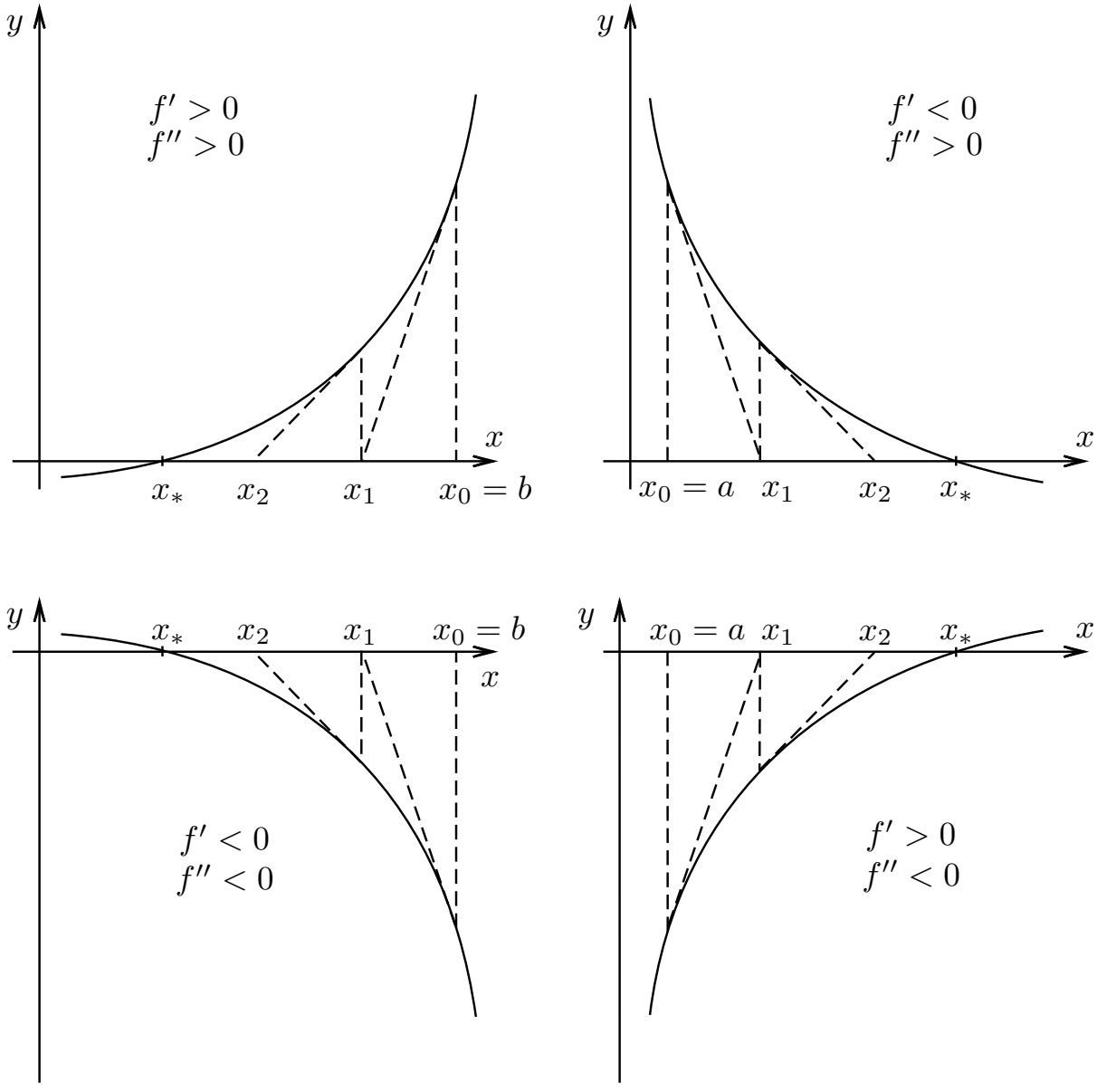


Рис. 2.3: Условия сходимости метода Ньютона

Иллюстрация теоремы приведена на рис. 2.3. Поскольку формулировки и доказательства всех пунктов теоремы совершенно аналогичны, ограничимся доказательством первого пункта.

Доказательство. Монотонность последовательности $\{x_k\}$ докажем по индукции. По условию $x_0 = b$. Предположим, что для некоторого $k \geqslant 0$ выполняются неравенства $x_* < x_k \leqslant b$, и докажем, что тогда

$$x_* < x_{k+1} < x_k. \tag{2.6}$$

Так как $f(x_*) = 0$ и $x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$, то справедливо $x_k - x_{k+1} = \frac{f(x_k)-f(x_*)}{f'(x_k)}$. Воспользуемся формулой конечных приращений Лагранжа. Тогда получим

$$x_k - x_{k+1} = \frac{(x_k - x_*)f'(\xi_k)}{f'(x_k)}, \text{ где } \xi_k \in (x_*, x_k). \tag{2.7}$$

Пусть выполнены условия $f'(x) > 0$ и $f''(x) > 0$. Тогда

$$0 < \frac{f'(\xi_k)}{f'(x_k)} < 1,$$

причём последнее неравенство является следствием монотонного возрастания $f'(x)$. Таким образом,

$$0 < \frac{(x_k - x_*)f'(\xi_k)}{f'(x_k)} < x_k - x_*,$$

и из (2.7) получим $0 < x_k - x_{k+1} < x_k - x_*$, т.е. получим требуемое неравенство (2.6). Таким образом, последовательность $\{x_k\}$ монотонно убывает и ограничена снизу числом x_*. Поэтому данная последовательность имеет предел, который в силу непрерывности функции $f(x)$ и условия $f'(x_*) \neq 0$ совпадает с корнем x_* уравнения $f(x) = 0$. □

Определение 2. *Число x_* является корнем уравнения $f(x) = 0$ кратности p, если $f(x_*) = f'(x_*) = \ldots = f^{(p-1)}(x_*) = 0$, но $f^{(p)}(x_*) \neq 0$.*

Корень кратности $p = 1$ называется *простым*.

Только простой корень метод Ньютона находит быстро. Поиск кратного корня сильно замедляется. Рассмотрим в качестве примера корень

кратности 2 (то есть $f(x_*) = f'(x_*) = 0$, но $f''(x_*) \neq 0$). Очевидно, в выражении (2.5) производная $f'(x_n)$ близка к 0. Разложим $f'(x_n)$ в ряд Тейлора с центром в точке x_*:

$$f'(x_n) = \underbrace{f'(x_*)}_{=0} + f''(\eta_n)(x_n - x_*) = f''(\eta_n)(x_n - x_*),$$

где $\eta_n \in [x_n, x_*]$. В итоге (2.5) можно переписать так:

$$|x_{n+1} - x_*| = \frac{1}{2}\frac{|f''(\xi_n)|}{|f'(x_n)|}|x_n - x_*|^2 = \frac{1}{2}\frac{|f''(\xi_n)|}{|f''(\eta_n)(x_n - x_*)|}|x_n - x_*|^2 = \\ = \frac{1}{2} \cdot \frac{|f''(\xi_n)|}{|f''(\eta_n)|} \cdot |x_n - x_*|.$$

У множителя $|x_n - x_*|$ пропала степень 2. То есть скорость убывания погрешности стала линейной.

В случае кратного корня применяют *модифицированный* метод Ньютона: $x_{n+1} = x_n - \frac{pf(x_n)}{f'(x_n)}$, где p — кратность корня. Добавление коэффициента p сохраняет квадратичную скорость сходимости к корню. Обоснование этого факта приведено в конце пособия в разделе «приложения».

Очевидным недостатком метода Ньютона является необходимость вычисления производной на каждой итерации. Может оказаться, что на подготовку очередного значения $f'(x_n)$ уйдет слишком много машинного времени и это перевесит выигрыш в малом числе итераций метода Ньютона.

Иногда упрощают вычисления, используя на каждой итерации значение производной в точке x_0 (заменяют $f'(x_n)$ на $f'(x_0)$): $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_0)}$. Это, к сожалению, лишает метод квадратичной скорости сходимости.

1.2.4 Метод секущих

Очевидно, вблизи корня касательная к графику функции в точке $(x_n, f(x_n))$ и прямая, проходящая через точки $(x_{n-1}, f(x_{n-1}))$ и $(x_n, f(x_n))$ очень близки. Можно приближённо считать

$$f'(x_n) \approx \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}.$$

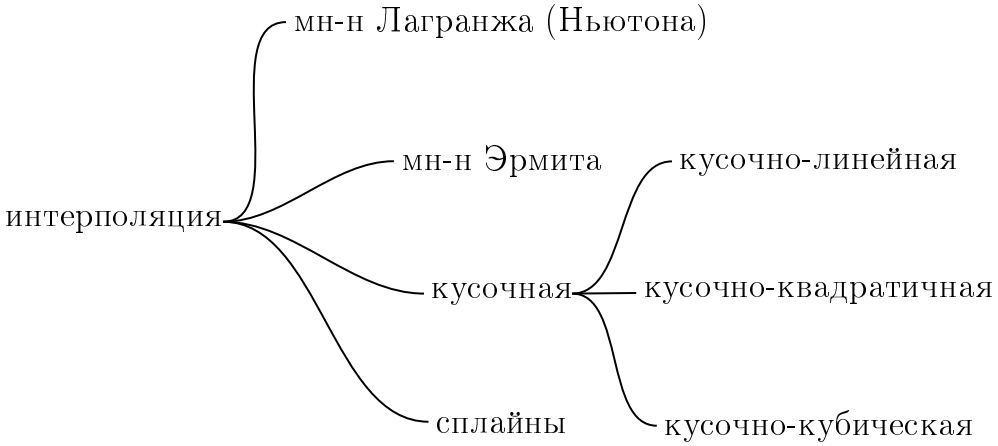
Последняя замена даёт метод *секущих*:

$$x_{n+1} = x_n - \frac{f(x_n)(x_n - x_{n-1})}{f(x_n) - f(x_{n-1})}.$$

Скорость убывания погрешности в методе секущих $q^{1{,}62n}$. Это выше линейной скорости q^n, но меньше квадратичной q^{2n}.

Метод	Дихотомия	Простые итерации	Ньютон	Секущие
Убывание ошибки	$(1/2)^n$	q^n, $0 < q < 1$	q^{2n}, $0 < q < 1$	$q^{1{,}62}$, $0 < q < 1$

1.3 Лекция 3



1.3.1 Интерполяция многочленами

Пусть задана конечная таблица $\begin{array}{c|c|c|c|c} x_0 & x_1 & x_2 & \cdots & x_n \\ \hline y_0 & y_1 & y_2 & \cdots & y_n \end{array}$, где $x_0 < x_1 < \ldots < x_n$, отражающая некоторую функциональную зависимость $y(x)$. Такая таблица может быть получена в ходе проведения эксперимента или в результате трудоёмких расчётов. Последнее означает, что получение значение $y(x)$, где x не содержится в таблице, может быть невозможно (например, если эксперимент уже закончен) или сопряжено с большими затратами (например, несколько минут или часов машинного или человеческого времени).

Но на практике нужно знать значение функции в точках отличных от табличных. Различают два случая: определение $y(x)$, где $x \in [x_0, x_n]$ (аргумент x находится между табличными значениями) — это *интерполяция*; и определение $y(x)$, где $x \notin [x_0, x_n]$ (аргумент x находится за пределами табличных значений) — это *экстраполяция*.

Одним из широко используемых способов приближения функции, заданной таблично в $n+1$ точке, есть приближение её многочленами степени n.

Многочлен Лагранжа. Будем называть x_i, $i = 0, 1, \ldots, n$ *узлами интерполяции*.

Рассмотрим многочлен $P_n(x) = a_n x^n + a_{n-1}x^{n-1} + \ldots + a_1 x + a_0$ степени не выше n (некоторые коэффициенты, включая главный коэффициент, могут равняться нулю). Потребуем, чтобы $P_n(x)$ совпадал с функцией в табличных точках, то есть пусть $P_n(x_k) = y_k$, где $k = 0, 1, \ldots, n$. Получим систему уравнений относительно a_i, $k = 0, 1, \ldots, n$:

$$\begin{cases} a_n x_0^n + a_{n-1}x_0^{n-1} + \ldots + a_1 x_0 + a_0 = y_0, \\ a_n x_1^n + a_{n-1}x_1^{n-1} + \ldots + a_1 x_1 + a_0 = y_1, \\ \ldots \\ a_n x_n^n + a_{n-1}x_n^{n-1} + \ldots + a_1 x_n + a_0 = y_n. \end{cases}$$

Определитель последней системы есть определитель Вандермонда (см. приложение):

$$\Delta = \begin{vmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ \vdots & \vdots & & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{vmatrix} = \prod_{\substack{i,j=0 \\ (i \neq j)}}^{n} (x_i - x_j).$$

В самом начале мы потребовали, чтобы $x_0 < x_1 < \ldots < x_n$. Значит, ни одна скобка не даст нуля и, следовательно, $\Delta \neq 0$. Последнее означает, что система имеет единственное решение. Несложно проверить, что следующий многочлен в точности соответствует этому решению (если подставить табличное x_k получим y_k):

$$P_n(x) = y_0 \frac{(x - x_1)(x - x_2)\ldots(x - x_n)}{(x_0 - x_1)(x_0 - x_2)\ldots(x_0 - x_n)} + \ldots + \\ + y_k \frac{(x - x_0)\ldots(x - x_{k-1})(x - x_{k+1})\ldots(x - x_n)}{(x_k - x_0)\ldots(x_k - x_{k-1})(x_k - x_{k+1})\ldots(x_k - x_n)} + \ldots + \\ y_n \frac{(x - x_0)(x - x_1)\ldots(x - x_{n-1})}{(x_n - x_0)(x_n - x_1)\ldots(x_n - x_{n-1})} = \sum_{k=0}^{n} y_k \frac{L_n^{(k)}(x)}{L_n^{(k)}(x_k)},$$

где $L_n^{(k)}(x) = (x - x_0)\ldots(x - x_{k-1})(x - x_{k+1})\ldots(x - x_n)$ — полиномы n-ой степени специального вида.

Многочлен Ньютона. Приведём ещё одну форму записи интерполяционного полинома:

$$P_n(x) = A_0 + A_1(x - x_0) + A_2(x - x_0)(x - x_1) + \ldots + \\ + A_n(x - x_0)(x - x_1)\ldots(x - x_{n-1}). \quad (3.1)$$

Требование совпадения значений полинома с заданными значениями функции приводит к системе линейных уравнений с *треугольной* матрицей для неопределённых коэффициентов A_i, $i = 0, 1, \ldots, n$:

$$\begin{cases} A_0 = f_0, \\ A_0 + A_1(x_1 - x_0) = f_1, \\ A_0 + A_1(x_2 - x_0) + A_2(x_2 - x_0)(x_2 - x_1) = f_2, \\ \ldots \\ A_0 + A_1(x_n - x_0) + A_2(x_n - x_0)(x_n - x_1) + \ldots + \\ \qquad + A_n(x_n - x_0)(x_n - x_1)\cdots(x_n - x_{n-1}) = f_n. \end{cases}$$

численное решение которой не составляет труда.

Интерполяционный полином, записанный в форме (3.1), называется полиномом *Ньютона*. Он интересен тем, что каждая частичная сумма его первых $(m+1)$ слагаемых представляет собой интерполяционный полином m-й степени, построенный по первым $(m + 1)$ табличным данным.

Сравнение форм Лагранжа и Ньютона. Многочлен Лагранжа и многочлен Ньютона суть один и тот же многочлен, записанный в различных формах. У каждой из форм записи есть свои достоинства:

Многочлен Лагранжа	*Многочлен Ньютона*
удобно, если требуется приближать различные функции, заданные табличными значениями в одних и тех же точках	• удобно, если в качестве результата нужна непосредственно формула, приближающая функцию $f(x)$ • удобно, если требуется добавить новый узел x_{n+1}; достаточно найти только новый неизвестный коэффициент A_{n+1}, остальные A_i, $i = 0, 1, \ldots, n$ остаются неизменными

Погрешность интерполяции. Ошибка приближения функции интерполяционным полиномом n-ой степени в точке x — это разность $R_n(x) = f(x) - P_n(x)$.

Рассмотрим полином специального вида $(n+1)$-ой степени:

$$\omega_{n+1}(x) = (x - x_0)(x - x_1)\ldots(x - x_n) = \prod_{i=0}^{n}(x - x_i).$$

Теорема 3.1. *Пусть на отрезке $[a, b]$, таком, что $x_i \in [a, b]$, $i = 0, 1, \ldots, n$, функция $f(x)$ $(n+1)$ раз непрерывно дифференцируема. Тогда*

$$R_n(x) = \frac{f^{(n+1)}(x')}{(n+1)!}\omega_{n+1}(x), \quad \text{где } x' \in [a, b].$$

Доказательство. Будем искать погрешность в виде

$$R_n(x) = C(x)\omega_{n+1}(x), \tag{3.2}$$

где $C(x)$ — функция, ограниченная на $[a, b]$ (при такой форме записи выражения для погрешности гарантируется, что она обращается в ноль в узлах интерполяции).

Чтобы получить представление о $C(x)$, рассмотрим вспомогательную функцию

$$\varphi(x) = f(x) - P_n(x) - C(\xi)\omega_{n+1}(x), \tag{3.3}$$

где ξ — некоторое фиксированное значение на отрезке $[a, b]$. Очевидно, на $[a, b]$ функция $\varphi(x)$ имеет $(n + 2)$ нуля. Это узлы интерполяции и точка $x = \xi$. Согласно теореме Ролля, существует точка $x' \in [a, b]$, в которой $\varphi^{(n+1)}(x') = 0$. Продифференцировав 3.3 $(n + 1)$ раз и подставив $x = x'$, получим

$$0 = \varphi^{(n+1)}(x') = f^{(n+1)}(x') - (n + 1)!C(\xi).$$

Отсюда $C(\xi) = \frac{f^{(n+1)}(x')}{(n+1)!}$. (Ясно, что x' в теореме Ролля зависит от расположения нулей функции $\varphi(x)$; тем самым x' представляет собой некоторую неявную зависимость $x' = x'(\xi)$ и полученное отношение действительно определяет функцию от ξ.)

Переобозначая $C(\xi)$ на $C(x)$ и учитывая 3.2, получаем утверждение теоремы. □

Хочется контролировать поведение $\omega_{n+1}(x)$ на отрезке $[a, b]$. Сравним, к примеру, поведение на отрезке $[-1, 1]$ $\omega_{10}(x)$ с выбором равноотстоящих узлов $(-1, -\frac{8}{9}, -\frac{7}{9}, \ldots, \frac{8}{9}, 1)$ и $\overline{\omega}_{10}(x)$ с узлами Чебышёва ($\cos\frac{19\pi}{20}$, $\cos\frac{17\pi}{20}$, $\cos\frac{15\pi}{20}, \ldots, \cos\frac{\pi}{20}$).

Из рисунка 3.1 видно, что многочлен $\overline{\omega}_{10}(x)$ меньше отклоняется от оси Ox: $\max\limits_{[-1,1]} \overline{\omega}_{10}(x) \approx 2 \cdot 10^{-3} < 13 \cdot 10^{-3} \approx \max\limits_{[-1,1]} \omega_{10}(x)$. Оказывается, что среди всех многочленов $\omega_{n+1}(x)$ с главным коэффициентом 1 и $(n + 1)$ корнем на отрезке $[a, b]$ многочлены Чебышёва менее всего отклоняются от нуля.

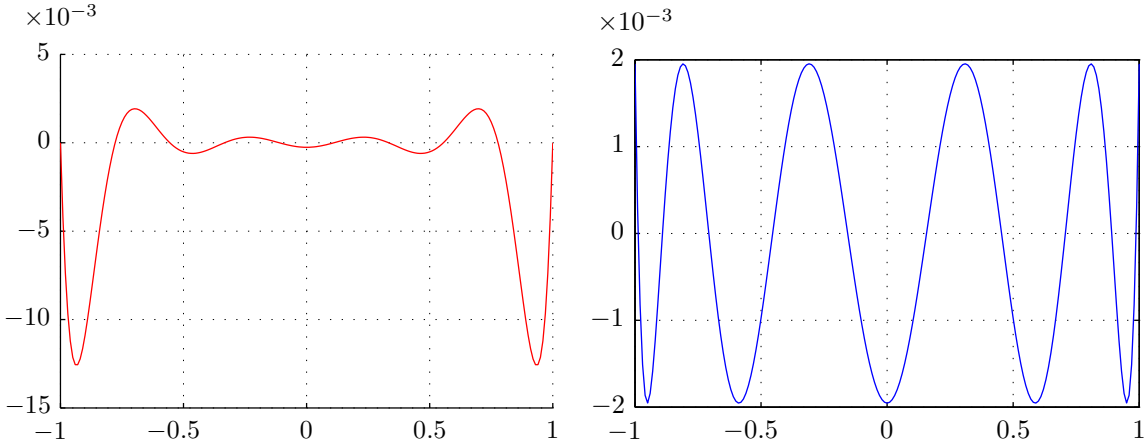
При больших $n > 10$ интерполяция по равноотстоящим узлам практически не используется:

1. может не быть сходимости (функция Рунге $f(x) = \frac{1}{1+25x^2}$, у которой ошибка интерполяции с ростом n бесконечно возрастает);

2. даже малые погрешности в табличных данных приводят к большим (неустранимым) ошибкам интерполяции.

При интерполяции по чебышёвским узлам этих неприятностей нет.

В таблице приведены результаты приближения интерполяционными полиномами различной степени функции Рунге



а) б)

Рис. 3.1: Многочлен $\omega_{n+1}(x) = (x - x_0)(x - x_1) \cdot \ldots \cdot (x - x_n)$ при выборе а) равноотстоящих на $[-1, 1]$ узлов; б) чебышёвских узлов.

n	$0,7 < x < 1$	$\lvert x\rvert < 0,7$	Чебышёвские узлы
4	0,44	0,37	0,40
8	1,01	0,24	0,17
10	1,88	0,3	0,11
20	40,0	0,12	0,01

Функция Рунге — «нехорошая» для интерполирования функция.

1.4 Лекция 4

1.4.1 Многочлены Чебышёва.

Рекуррентная форма записи Многочлены Чебышёва $T_n(x)$, где $n \geqslant 0$, определяются соотношениями

$$\begin{aligned} &T_0(x) = 1, \qquad T_1(x) = x, \\ &T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x) \text{ при } n > 0. \end{aligned} \tag{4.1}$$

Например,

$$\begin{aligned} &T_2(x) = 2x^2 - 1, \qquad &&T_3(x) = 4x^3 - 3x, \\ &T_4(x) = 8x^4 - 8x^2 + 1, \quad &&T_5(x) = 16x^5 - 20x^3 + 5x. \end{aligned}$$

Тригонометрическая форма записи. Для любого θ справедливо $\cos((n+1)\theta) = 2\cos\theta\cos n\theta - \cos((n-1)\theta)$. При $\theta = \arccos x$ получим

$$\cos((n+1)\arccos x) = 2x\cos(n\arccos x) - \cos((n-1)\arccos x). \tag{4.2}$$

Рассмотрим выражение $\cos(n\arccos x)$ при $n = 0$ и $n = 1$:

$$\cos(0 \cdot \arccos x) = 1 = T_0(x), \qquad \cos(1 \cdot \arccos x) = x = T_1(x). \tag{4.3}$$

Видно, что (4.3) и (4.2) равносильно (4.1), поэтому при всех n $T_n(x) = \cos(n\arccos x)$.

Явная форма записи. Рекуррентное соотношение (4.1) является разностным. Для его решения заменяют $T_n(x)$ на μ^n. После подстановки и сокращения получаем:

$$\mu^2 - 2\mu x + 1 = 0$$

с корнями

$$\mu_{1,2} = x \pm \sqrt{x^2 - 1}.$$

При $x \neq \pm 1$ корни простые, поэтому

$$T_n(x) = c_1(x)\mu_1^n + c_2(x)\mu_2^n.$$

Из системы

$$\begin{cases} T_0(x) = 1 = c_1(x) + c_2(x), \\ T_1(x) = x = c_1(x)(x + \sqrt{x^2-1}) + c_2(x)(x - \sqrt{x^2-1}). \end{cases}$$

следует, что $c_1 = c_2 = 1/2$. Таким образом,

$$T_n(x) = \frac{(x + \sqrt{(x^2-1)})^n + (x - \sqrt{(x^2-1)})^n}{2}.$$

Свойства:

1. $T_{2n}(x)$ — чётные функции, $T_{2n+1}(x)$ — нечётные функции.
2. $T_n(x)$ выражается через косинус, следовательно $|T_n(x)| \leqslant 1$ при $x \in [-1, 1]$.
3. Из уравнения $T_n(x) = \cos(n \arccos x) = 0$ получаем, что
$$x_k = \cos\left(\frac{(2k-1)\pi}{2n}\right), \quad k = 1, \ldots, n$$
— нули $T_n(x)$.
4. Из уравнения $T_n'(x) = -\sin(n \arccos x)\frac{-n}{\sqrt{1-x^2}} = 0$ получаем, что
$$\xi_k = \cos\left(\frac{k\pi}{n}\right), \quad k = 0, \ldots, n$$
— точки экстремума $T_n(x)$. Заметим, что $T_n(\xi_k) = (-1)^k$.

Геометрическая интерпретация. Если верхнюю полуокружность единичного радиуса разделить на n частей, то середины дуг — координаты нулей, экстремумы — точки деления (рис. 4.1).

Наименьшее отклонение от нуля Так как $T_0(x) = 1$ и $T_{n+1}(x) = 2xT_n(x) - \ldots$, то коэффициент при главном члене равен 2^{n-1}. В погрешности многочлена Лагранжа участвует многочлен $\omega_{n+1} = (x-x_0)\cdot\ldots\cdot(x-x_n)$ с старшим коэффициентом 1. Поэтому рассматривают также изменённый многочлен Чебышёва $\overline{T}_x(n) = 2^{1-n}T_n(x)$ со старшим коэффициентом 1. Справедлива следующая

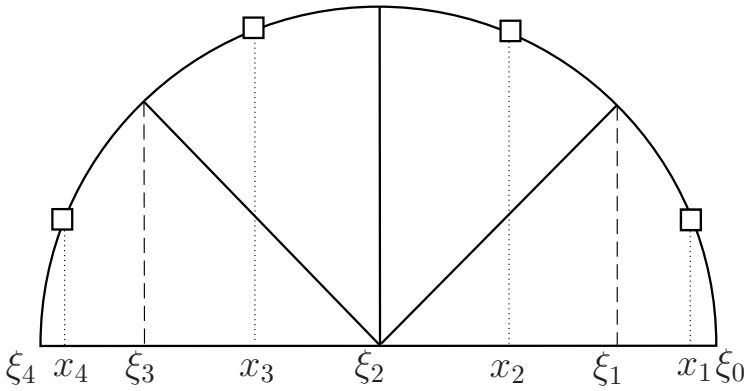


Рис. 4.1: Геометрическая интерпретация корней и точек экстремума полинома Чебышёва при $n = 4$.

Теорема 4.1. *Для всякого многочлена $P_n(x)$ степени n с единичным старшим коэффициентом имеет место неравенство*

$$\max_{x\in[-1,1]} |P_n(x)| \geqslant \max_{x\in[-1,1]} |\overline{T}_n(x)| = 2^{1-n},$$

причём знак равенства возможен только в случае $P_n(x) = \overline{T}_n(x)$.

Доказательство «$\geqslant$». Будем действовать от противного. Пусть найдётся такой многочлен, что

$$\max_{x\in[-1,1]} |P_n(x)| < \max_{x\in[-1,1]} |\overline{T}_n(x)| = 2^{1-n}. \tag{4.4}$$

Рассмотрим многочлен $Q_{n-1}(x) = \overline{T}_n(x) - P_n(x)$ степени не выше $n-1$ (оба слагаемых имеют старший единичный коэффициент). Подставим в него точки $\xi_n^k = \sin\frac{k\pi}{n}$, $k = 0, \ldots, n$ (точки экстремума многочлена $\overline{T}_n(x)$):

$$\operatorname{sign}(Q_{n-1}(\xi_n^i)) = \operatorname{sign}((-1)^k 2^{1-n} - P_n(\xi_n^k)) \overset{\text{из-за (4.4)}}{=} \\ = \operatorname{sign}((-1)^k 2^{1-n}) = (-1)^k.$$

Заметим, что знак многочлена $Q_{n-1}(x)$ меняется $n+1$ раз на отрезке $[-1, 1]$ (т.к. $k = 0, \ldots, n$). Значит, многочлен $Q_{n-1}(x)$ степени не выше $n-1$ имеет n различных корней. Получили противоречие. □

Доказательство единственности. □

Из-за последней теоремы многочлен $\overline{T}_n(x)$ получил название *наименее уклоняющегося от нуля.*

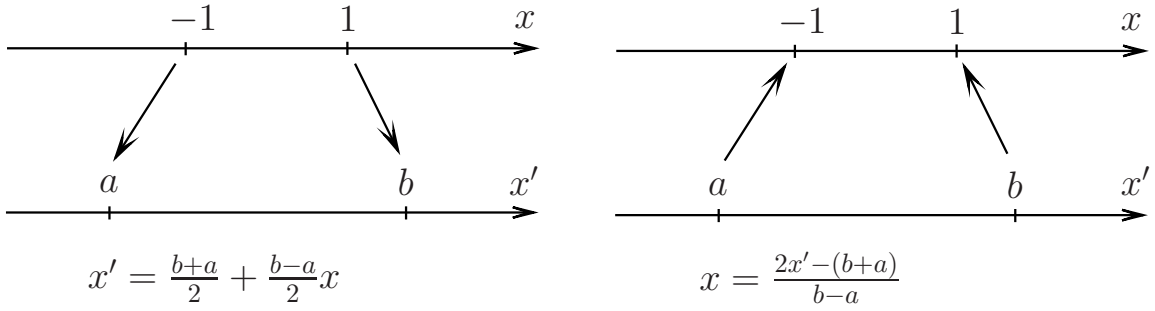


Рис. 4.2: Линейные преобразования отрезка $[-1,1]$ в $[a,b]$ и обратно

Заметим, что теорема работает только на отрезке $[-1,1]$. Хочется снять это ограничение. Для этого рассматривают линейные преобразования отрезка $[-1,1]$ в $[a,b]$ $x' = \frac{b+a}{2} + \frac{b-a}{2}x$ и обратно $x = \frac{2x'-(b+a)}{b-a}$. Получаем многочлен

$$\overline{T}_n^{[a,b]}(x) = \left(\frac{b-a}{2}\right)^n \overline{T}_n\left(\frac{2x-(b+a)}{b-a}\right) = (b-a)^n 2^{1-2n} T_n\left(\frac{2x-(b+a)}{b-a}\right)$$

со старшим коэффициентом 1, наименее уклоняющийся от нуля на отрезке $[a,b]$. Будем называть $\overline{T}_n^{[a,b]}(x)$ также *чебышёвским*. Нетрудно проверить, что нулями многочлена $\overline{T}_n^{[a,b]}(x)$ являются точки

$$x_k = \frac{b+a}{2} + \frac{b-a}{2}\cos\left(\frac{(2k-1)\pi}{2n}\right), \quad k = 1,\ldots,n. \tag{4.5}$$

Теперь можно дать ответ на вопрос, как уменьшить погрешность интерполяции $R_n(x)$ за счёт выбора узлов интерполяции. Если в качестве узлов интерполяции выбрать корни (4.5) многочлена $\overline{T}_{n+1}^{[a,b]}(x)$, тогда

$$\max_{a\leqslant x\leqslant b} |\omega_{n+1}(x)| = \max_{a\leqslant x\leqslant b} |\overline{T}_{n+1}^{[a,b]}(x)| = (b-a)^{n+1}2^{1-2(n+1)}.$$

При этом улучшить (т.е. уменьшить) последнюю величину уже нельзя. Получаем $|R_n(x)| \leqslant \frac{M_{n+1}}{(n+1)!}(b-a)^{n+1}2^{1-2(n+1)}$, где $M_{n+1} = \max\limits_{a\leqslant x\leqslant b} |f^{(n+1)}(x)|$.

1.4.2 Среднеквадратическое приближение (метод наименьших квадратов)

Рассмотрим принципиально иной способ приближения функций, заданных таблицей своих значений $\begin{array}{c|c|c|c} x_0 & x_1 & \cdots & x_n \\ \hline y_0 & y_1 & \cdots & y_n \end{array}$. Будем искать приближение в виде полинома степени m: $P_m(x) = a_0 + a_1 x + \ldots + a_m x^m$, такого,

который минимизирует сумму квадратов отклонений полинома от заданных значений функции:

$$\Phi(a_0, a_1, \ldots, a_m) = \sum_{i=0}^{n} (P_m(x_i) - y_i)^2.$$

Ясно, что при $m = n$ решением задачи является полином Лагранжа, поскольку на нём достигается абсолютный минимум: $\Phi = 0$. Известно, что при $m < n$ задача имеет единственное решение. При $m > n$ задача имеет бесконечное множество решений.

Рассмотрим случай $m < n$. Условия минимума функции Φ следуют из математического анализа:

$$\frac{\partial \Phi}{\partial a_k} = 2 \sum_{i=0}^{n} (P_m(x_i) - y_i) x_i^k = 0, \qquad k = 0, 1, ..., m.$$

После подстановки выражения для $P_n(x)$ и перегруппировки слагаемых, получим:

$$a_0 \sum_{i=0}^{n} x_i^{k+0} + a_1 \sum_{i=0}^{n} x_i^{k+1} + \ldots + a_m \sum_{i=0}^{n} x_i^{k+m} = \sum_{i=0}^{n} y_i x_i^k, \qquad k = 0, \ldots, m.$$

Эта система линейных уравнений с симметричной матрицей:

$$\begin{pmatrix} n+1 & \sum x_i & \cdots & \sum x_i^m \\ \sum x_i & \sum x_i^2 & \cdots & \sum x_i^{m+1} \\ \sum x_i^2 & \sum x_i^3 & \cdots & \sum x_i^{m+2} \\ \vdots & \vdots & \ddots & \vdots \\ \sum x_i^m & \sum x_i^{m+1} & \cdots & \sum x_i^{m+m} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_m \end{pmatrix} = \begin{pmatrix} \sum y_i \\ \sum y_i x_i \\ \sum y_i x_i^2 \\ \vdots \\ \sum y_i x_i^m \end{pmatrix}.$$

Полином степени $m < n$ с коэффициентами, найденными таким образом, называется среднеквадратичным приближением функции, заданной таблицей. (Или наилучшим среди полиномов степени m приближением к функции по табличным данным.)

Соответствующую погрешность приближения можно характеризовать среднеквадратичным отклонением $\Delta = \frac{1}{n+1} \sum_{i=0}^{n} [P_m(x_i) - y_i]^2$.

Основная сфера применения — обработка экспериментальных данных.

Экспериментальные данные характеризуются значительным разбросом (ошибки измерения, экспериментальный «шум» и т.д.) Интерполяционный полином, построенный по этим точкам, плохо отражает поведение функции $f(x)$. Среднеквадратичный полином «сглаживает шум».

Пример. Пусть известно, что величина y является некоторой функцией от аргумента x, причём в результате измерений получена таблица значений $y_k = y(x_k)$, $k = 1, 2, 3, 4$.

Полученные измерения позволяют приближённо считать, что зависимость $y = y(x)$ является линейной, т.е.

$$y = ax + b, \tag{4.6}$$

где a, b - некоторые числа. Числа a, b в эмпирической формуле (4.6) необходимо подобрать таким образом, чтобы при значениях $x = x_k$ $(k = 1, 2, 3, 4)$ выполнялись условия:

$$ax_1 + b = y_1, \quad ax_2 + b = y_2, \quad ax_3 + b = y_3, \quad ax_4 + b = y_4. \tag{4.7}$$

Получилась система четырёх линейных уравнений относительно двух неизвестных a, b. Классического решения данной системы нет.

Введем функцию $\Phi(a, b) = \sum\limits_{k=1}^{4} (ax_k + b - y_k)^2$, равную сумме квадратов невязок, и примем за обобщённое решение системы (4.7) ту пару чисел (a, b), для которой функция $\Phi(a, b)$ принимает наименьшее значение. Получим систему двух уравнений:

$$\overbrace{\frac{\partial \Phi}{\partial a} = 0, \quad \frac{\partial \Phi}{\partial b} = 0.}$$

Данная система имеет обычное классическое решение.

1.4.3 Многочлены Эрмита

Предположим, что функция задана конечным набором своих значений, а также некоторых производных (возможно, не во всех точках). В таблице, K_i определяет количество данных в i-ом узле. Например, если в узле x_i

заданы значение функции y_i и производной y'_i, то $K_i = 2$. Для всякого $i = 0, \ldots, n$ $K_i \geqslant 1$, т.е. в каждой колонке обязательно присутствует y_i — значение функции в точке x_i. Обозначим $p = K_1 + K_2 + \ldots + K_n - 1$. Многочлен $H_p(x)$ степени p называется многочленом Эрмита, если

$$H_p^{(k)}(x) = y_i^{(k)}, \quad i = 0, \ldots, n, \quad k = 0, 1, \ldots, K_i - 1.$$

x	x_0	x_1	$\cdots$	x_n
y	y_0	y_1	$\cdots$	y_n
y'	y'_0	y'_1	$\cdots$	y'_n
y''	$\vdots$	$\vdots$	$\cdots$	$\vdots$
y'''				
$\vdots$	$y_0^{(K_0-1)}$	$y_0^{(K_1-1)}$	$\cdots$	$y_0^{(K_n-1)}$
	K_0	K_1	$\cdots$	K_n

Погрешность для многочлена Эрмита выражается аналогично погрешности для многочлена Лагранжа. Если интерполяция происходит на отрезке $[a, b]$, содержащем x_i, $i = 0, \ldots, n$ и функция $f(x)$ $(p+1)$ раз непрерывно дифференцируема, то погрешность выражается формулой:

$$R_p(x) = f(x) - H_p(x) = \frac{f^{(p+1)}(\xi)}{(p+1)!}(x - x_0)^{K_0}(x - x_1)^{K_1} \ldots (x - x_n)^{K_n},$$

где ξ неизвестная точка принадлежащая интервалу $[a, b]$. Принято обозначать $\omega_{p+1}(x) = (x - x_0)^{K_0}(x - x_1)^{K_1} \ldots (x - x_n)^{K_n}$.

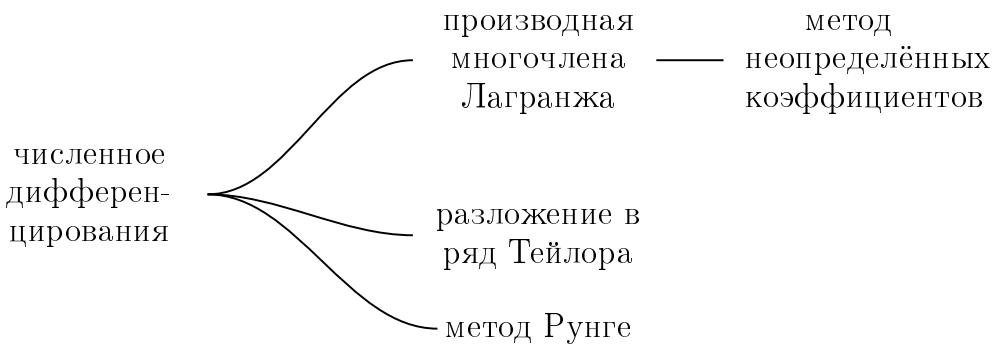
1.4.4 Интерполяция кубическими сплайнами

Сплайном, соответствующим данной функции $f(x)$ и данным узлам $x_0, \ldots x_n$, называется функция $s(x)$, удовлетворяющая следующим условиям:

1. на каждом сегменте $[x_{i-1}, x_i]$, $i = 1, 2, \ldots, n$,функция $s(x)$ является многочленом третьей степени;
2. функция $s(x)$, а также её первая и вторая производные непрерывны на $[a, b]$;
3. $s(x_i) = f(x_i)$, $i = 0, 1, \ldots, n$.

$$\begin{cases} s_i(x_i) = f(x_i), \\ s_i'(x_i) = s_{i-1}'(x_i), \\ s_i''(x_i) = s_{i-1}''(x_i). \end{cases}$$

1.5 Лекция 5



1.5.1 Кусочная интерполяция (КИ)

Оценка погрешности интерполяции функции $f(x)$ с помощью ИП $P_n(x)$ составляет:

$$|R_n(x)| = |f(x) - P_n(x)| \leqslant \frac{M_{n+1}}{(n+1)!} |\omega_{n+1}(x)|,$$

где $M_{n+1} = \max\limits_{[a,b]} |f^{(n+1)}(x)|$ — получена в предположении существования $(n+1)$-ой непрерывной производной функции $f(x)$.

На практике же далеко не всегда приходится иметь дело с очень гладкими функциями (у которых первая и высшие производные непрерывны). В связи с этим часто применяется *кусочная* интерполяция. Для приближения функции в точке x строится полином невысокой степени по данным в табличных точках, ближайшим к точке x.

Пусть необходимо вычислить $f(x)$ для $x \in [x_i, x_{i+1}]$.

Кусочно-линейная интерполяция. Для вычисления используется линейное приближение

$$f(x) \approx f_i + \frac{f_{i+1} - f_i}{h}(x - x_i), \text{ где } h = x_{i+1} - x_i.$$

Очевидно, что при подстановке $x = x_i$ или $x = x_{i+1}$ получается тождество.

Кусочно-квадратичная интерполяция. Привлекается ещё одна табличная точка (x_{i-1} или x_{i+1}), и строится полином второй степени:

$$f(x) \approx f_i + \frac{f_{i+1} - f_i}{h}(x - x_i) + \frac{f_{i+1} - 2f_i + f_{i-1}}{2h^2}(x - x_i)(x - x_{i+1}).$$

Очевидно, что при подстановке $x = x_i$ или $x = x_{i+1}$ или $x = x_{i-1}$ получается тождество.

Кусочно-кубическая интерполяция.

$$f(x) \approx f_i + \frac{f_{i+1} - f_i}{h}(x - x_i) + \frac{f_{i+1} - 2f_i + f_{i-1}}{2h^2}(x - x_i)(x - x_{i+1}) + \\ + \frac{f_{i+2} - 3f_{i+1} + 3f_i - f_{i-1}}{6h^3}(x - x_{i-1})(x - x_i)(x - x_{i+1}).$$

Нетрудно убедиться, что при $f_{i-1}, f_i, f_{i+1}, f_{i+2}$ в последних двух формулах стоят биномиальные коэффициенты.

Более высокие степени ИП для КИ, как правило, не используют. Узлы интерполяции при кусочной интерполяции берутся вблизи интересующего нас узла x_i.

1.5.2 Другие способы интерполяции

По аналогии с интерполяционным многочленом $P_n(x) = a_n x^n + a_{n-1}x^{n-1} + \ldots + a_0$ можно искать приближение таблично заданной функции в виде:

$$P_n(x) = a_n\varphi_n(x) + a_{n-1}\varphi_{n-1}(x) + \ldots + a_0\varphi_0(x)$$

по системе линейно независимых функций $\{\varphi_k(x) :\ k = 0, 1, \ldots, n\}$.

Исходя из условий интерполяции (совпадения значения полинома с табличными значениями функции), для неопределённых коэффициентов $\{a_i\}$ получим систему линейных уравнений:

$$a_n\varphi_n(x_0) + a_{n-1}\varphi_{n-1}(x_0) + \ldots + a_0\varphi_0(x_0) = y_0,\ i = 0, 1, \ldots, n.$$

Для существования и единственности решения необходимо, чтобы детерминант удовлетворял условию

$$\Delta = \begin{vmatrix} \varphi_0(x_0) & \varphi_1(x_0) & \varphi_3(x_0) & \cdots & \varphi_n(x_0) \\ \varphi_0(x_1) & \varphi_1(x_1) & \varphi_3(x_1) & \cdots & \varphi_n(x_1) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \varphi_0(x_n) & \varphi_1(x_n) & \varphi_3(x_n) & \cdots & \varphi_n(x_n) \end{vmatrix} \neq 0.$$

Например, периодическую функцию может оказаться удобно приближать в виде полинома по системе функций $\{1, \sin(kx), \cos(kx)\}$ — это тригоно-

метрическая интерполяция. Часто используется, если функция разложима в ряд Фурье.

1.5.3 Обратная интерполяция

Пусть необходимо определить, при каком значении x функция $f(x)$ принимает заданное значение. Если функция $y(x)$ *строго монотонна* на интервале интерполирования $[a, b]$, то ничто не мешает поменять ролями x_i и y_i в таблице:

x_0	x_1	x_2	$\cdots$	x_n
y_0	y_1	y_2	$\cdots$	y_n

$\rightarrow$

y_0	y_1	y_2	$\cdots$	y_n
x_0	x_1	x_2	$\cdots$	x_n

С формальной точки зрения безразлично, что считать функцией и что аргументом. В этом случае представляется удобным приближать с помощью интерполяционного полинома зависимость $x = x(y)$. Это и есть *обратная* интерполяция. Многочлен в форме Лагранжа и в форме Ньютона выписывается стандартным образом.

Замечание. *Если допустить, что функция $y(x)$ строго монотонна на интервале интерполирования $[a, b]$, то можно получить ещё один способ решения нелинейных уравнений вида $y(x) = 0$. Пусть построен многочлен обратной интерполяции $\hat{P}_n(y)$ для обратной функции $x(y)$. Достаточно подставить в него ноль. Это даст корень уравнения $x_* = \hat{P}_n(0)$.*

1.5.4 Численное дифференцирование (ЧД) с помощью многочлена Лагранжа.

Приблизим функцию $f(x)$, заданную таблично, интерполяционным многочленом Лагранжа $P_n(x)$. В качестве значения $f'(x)$ приближённо можно принять $P'_n(x)$. Оценим возникающую погрешность. Известно, что

$$f(x) - P_n(x) = R_n(x) = \frac{f^{(n+1)}(\xi(x))}{(n+1)!}\omega_{n+1}(x),\ \xi(x) \in [a, b].$$

Отсюда

$$f'(x) - P_n'(x) = R_n'(x) = \frac{d}{dx}\left[\frac{f^{(n+1)}(\xi(x))}{(n+1)!}\omega_{n+1}(x)\right] =$$
$$= \frac{1}{(n+1)!}\left[f^{(n+2)}(\xi(x))\xi'(x)\omega_{n+1}(x) + f^{(n+1)}(\xi(x))\omega_{n+1}'(x)\right].$$

В записи присутствует производная неизвестной нам функции $\xi(x)$. Это мешает оценить $R_n'(x)$. При этом значение $R_n'(x)$ может оказаться достаточно большим (большим требуемой точности). Таким образом применение многочлена Лагранжа для численного нахождения производных в общем случае неприменимо.

1.5.5 Метод неопределённых коэффициентов.

Искомое выражение для производной k-то порядка в некоторой точке $x = x_i$ представляется в виде линейной комбинации заданных значений функции в узлах $x_0, x_1, \ldots, x_n$:

$$y^{(k)}(x_i) \approx c_0 y_0 + c_1 y_1 + \ldots + c_n y_n. \tag{5.1}$$

Предполагается, что это соотношение выполняется *точно*, если функция y является многочленом степени не выше n, т. е. может быть представлена в виде

$$y = b_0 + b_1(x - x_0) + \ldots + b_n(x - x_0)^n.$$

Отсюда следует, что соотношение (5.1), в частности, должно выполняться точно для многочленов $y = 1, y = x - x_0, \ldots, y = (x - x_0)^n$. Подставляя последовательно эти выражения в (5.1) и требуя выполнения точного равенства, получаем систему $n+1$ линейных алгебраических уравнений для определения неизвестных коэффициентов $c_0, c_1, \ldots, c_n$.

Пример. Найти выражение для производной y_1' в случае четырёх равноотстоящих узлов ($n = 3$ и $x_{i+1} - x_i = h = \text{const}$, где $i = 1, 2, 3$).

Приближение (3.10) запишется в виде

$$y_1' \approx c_0 y_0 + c_1 y_1 + c_2 y_2 + c_3 y_3. \tag{5.2}$$

Используем следующие многочлены:

$$y = 1, \quad y = x - x_0, \quad y = (x - x_0)^2, \quad y = (x - x_0)^3. \tag{5.3}$$

Вычислим их производные:

$$y' = 0, \quad y' = 1, \quad y' = 2(x - x_0), \quad y' = 3(x - x_0)^2. \tag{5.4}$$

Подставляем последовательно соотношения (5.3) и (5.4) соответственно в правую и левую части (5.2), при $x = x_1$ требуя выполнения точного равенства:

$$\begin{aligned}
0 = &\ c_0 \cdot 1 + c_1 \cdot 1 + c_2 \cdot 1 + c_3 \cdot 1, \\
1 = &\ c_0(x_0 - x_0) + c_1(x_1 - x_0)1 + c_2(x_2 - x_0)1 + c_3(x_3 - x_0), \\
2(x_1 - x_0) = &\ c_0(x_0 - x_0)^2 + c_1(x_1 - x_0)^2 1 + c_2(x_2 - x_0)^2 1 + c_3(x_3 - x_0)^2, \\
3(x_1 - x_0)^2 = &\ c_0(x_0 - x_0)^3 + c_1(x_1 - x_0)^3 1 + c_2(x_2 - x_0)^3 1 + c_3(x_3 - x_0)^3.
\end{aligned}$$

Получаем окончательно систему уравнений в виде

$$\begin{aligned}
c_0 + c_1 + c_2 + c_3 &= 0, \\
hc_1 + 2hc_2 + 3hc_3 &= 1, \\
hc_1 + 4hc_2 + 9hc_3 &= 2, \\
hc_1 + 8hc_2 + 27hc_3 &= 3.
\end{aligned}$$

Решая эту систему, получаем

$$c_0 = -\frac{1}{3h}, \quad c_1 = -\frac{1}{2h}, \quad c_2 = \frac{1}{h}, \quad c_3 = -\frac{1}{6h}.$$

Подставляя эти значения в (5.2), находим выражение для производной:

$$y_1' \approx \frac{1}{6h}(-2y_0 - 3y_1 + 6y_2 - y_3).$$

1.5.6 Разложение в ряд Тейлора.

Будем считать, что в таблице используется равномерный шаг: $x_i - x_{i-1} = h = \text{const}$, где $i = 1, 2, \ldots, n$. Разложим $f(x_{i-1})$ в ряд Тейлора в окрестности точки x_i:

$$f(x_{i-1}) = f(x_i) + (x_{i-1} - x_i)f'(x_i) + \frac{(x_{i-1} - x_i)^2}{2!}f''(\xi), \text{ где } x_{i-1} \leqslant \xi \leqslant x_i.$$

Для краткости заменим $f(x_i) \to f_i$ и $x_i - x_{i-1} \to h$:

$$f_{i-1} = f_i - hf_i' + h^2 f''(\xi)/2.$$

Откуда $f_i' = \frac{f_i - f_{i-1}}{h} + h\frac{f''(\xi)}{2}$. Приближённо можно положить $f_i' \approx \frac{f_i - f_{i-1}}{h}$. Ясно, что ошибка при этом будет не более, чем $hM_2/2 = O(h)$, где $M_2 = \max\limits_{[x_{i-1}, x_i]} |f''(x)|$. Полученную формулу для f_i' принято называть *левой разностью*.

Аналогично получается *правая разность*: $f_i' \approx \frac{f_{i+1} - f_i}{h} + O(h)$.

Разложим теперь f_{i-1} и f_{i+1} до h^3:

$$f_{i-1} = f_i - hf_i' + h^2\frac{f''(x)}{2!} - h^3\frac{f'''(\xi^-)}{3!},\ x_{i-1} \leqslant \xi^- \leqslant x_i,$$

$$f_{i+1} = f_i + hf_i' + h^2\frac{f''(x)}{2!} + h^3\frac{f'''(\xi^+)}{3!},\ x_i \leqslant \xi^+ \leqslant x_{i+1}.$$

Найдём разность последних двух равенств и выделим f_i':

$$f_i' = \frac{f_{i+1} - f_{i-1}}{2h} + h^2\frac{f'''(\xi^-) + f'''(\xi^+)}{6}.$$

Приближённое равенство $f_i' \approx \frac{f_{i+1} - f_{i-1}}{2h}$ называют *центральной разностью*. Погрешность при этом равна $f_i' - \frac{f_{i+1} - f_{i-1}}{2h} = h^2\frac{f'''(\xi^-) + f'''(\xi^+)}{6} = h^2\frac{f'''(\eta)}{3} = O(h^2)$[1], где $\eta \in [x_{i-1}, x_{i+1}]$.

Чтобы получить формулу для f_i'', разложим f_{i-1} и f_{i+1} до h^4

$$f_{i-1} = f_i - hf_i' + h^2\frac{f''(x)}{2!} - h^3\frac{f'''(x)}{3!} + h^4\frac{f^{\mathrm{IV}}(\zeta^-)}{4!},\ x_{i-1} \leqslant \zeta^- \leqslant x_i,$$

$$f_{i+1} = f_i + hf_i' + h^2\frac{f''(x)}{2!} + h^3\frac{f'''(x)}{3!} + h^4\frac{f^{\mathrm{IV}}(\zeta^+)}{4!},\ x_i \leqslant \zeta^+ \leqslant x_{i+1}.$$

Если сложить разложения f_{i-1} и f_{i+1} и затем выразить $f''(x)$, получим

$$f_i'' = \frac{f_{i+1} - 2f_i + f_{i-1}}{h^2} - h^2\frac{f^{\mathrm{IV}}(\zeta^-) + f^{\mathrm{IV}}(\zeta^+)}{24}.$$

Погрешность $f_i'' - \frac{f_{i+1} - 2f_i + f_{i-1}}{h^2} = -h^2\frac{f^{\mathrm{IV}}(\zeta^-) + f^{\mathrm{IV}}(\zeta^+)}{24} = -h^2\frac{f^{\mathrm{IV}}(\tau)}{12} = O(h^2)$, где $\tau \in [x_{i-1}, x_{i+1}]$, имеет порядок 2.

[1] Мы предполагаем, что $f'''(x)$ непрерывна на $[x_{i-1}, x_{i+1}]$. Пусть для определённости $f'''(\xi^-) \leqslant f'''(\xi^+)$. Тогда $f'''(\xi^-) \leqslant \frac{f'''(\xi^-) + f'''(\xi^+)}{2} \leqslant f'''(\xi^+)$. С силу непрерывности $f'''(x)$ найдётся точка $\eta \in [x_{i-1}, x_{i+1}]$ такая, что $f'''(\eta) = \frac{f'''(\xi^-) + f'''(\xi^+)}{2}$

Замечание. *Погрешность $O(h)$ говорит о том, что ошибка убывает пропорционально шагу h. Погрешность $O(h^2)$ говорит о том, что ошибка убывает пропорционально квадрату шага h. Например,*

Шаг	*Формула с погрешн. $O(h)$*	*Формула с погрешн. $O(h^2)$*
h	*погрешность* ε	*погрешность* ε
$h/2$	*погрешность* $\varepsilon/2$	*погрешность* $\varepsilon/4$
$h/4$	*погрешность* $\varepsilon/4$	*погрешность* $\varepsilon/16$

Про погрешность $O(h^p)$ говорят, что она имеет порядок p.

1.5.7 Неустойчивость формул численного дифференцирования

Рассмотрим влияние погрешности входных данных на результат вычисления производных по формулам ЧД. Пусть в точках x_i, $i = 0, 1, \ldots, n$ заданы значения функции $\tilde{y}_i$, которые отличаются от точных значений $y_i = y(x_i)$, т.е. $\tilde{y}_i = y_i \pm \delta_i$, где δ_i — погрешность входных данных. Величина $\delta = \max\limits_i \delta_i$ обычно бывает известна. Пусть в точке $x = x_i$ нужно приближённо вычислить $y'(x)$.

$$y'(x_i) \approx \frac{\tilde{y}_{i+1} - \tilde{y}_i}{h}.$$

Погрешность формулы

$$\Delta = \left| y' - \frac{\tilde{y}_{i+1} - \tilde{y}_i}{h} \right| = \left| \left(y' - \frac{y_{i+1} - y_i}{h} \right) + \left(\frac{y_{i+1} - y_i}{h} - \frac{\tilde{y}_{i+1} - \tilde{y}_i}{h} \right) \right| \leqslant$$
$$\leqslant \left| y' - \frac{y_{i+1} - y_i}{h} \right| + \left| \frac{y_{i+1} - \tilde{y}_{i+1}}{h} \right| + \left| \frac{y_i - \tilde{y}_i}{h} \right| \leqslant \frac{M_2}{2} h + \frac{\delta}{h} + \frac{\delta}{h} = \Phi(h),$$

где $M_2 = \max\limits_{[x_i, x_{i+1}]} |y''(x)|$. Здесь $\frac{M_2}{2} h$ — методическая ошибка, $\frac{2\delta}{h}$ — неустранимая погрешность. Наша цель — минимизировать ошибку, т.е. $\Phi(h)$. Для этого нельзя неограниченно уменьшать шаг h, т.к. $\Phi(h)$ в какой-то момент начинает расти. Найдём оптимальное значение h_*.

$$h_{opt}: \quad \Phi'(x) = \frac{M_2}{2} - \frac{2\delta}{h^2} = 0 \quad \Rightarrow \quad h_{opt} = 2\sqrt{\delta / M_2}$$

$$\Phi(h_{opt}) = \frac{M_2}{2} \cdot 2\sqrt{\delta / M_2} + 2\delta \cdot \frac{1}{2}\sqrt{M_2 / \delta} = 2\sqrt{M_2 \delta}.$$

Пример. $M_2 \sim 1,\ \delta \sim 0,01 \quad \Rightarrow \quad \Delta \sim 0,1.$

1.5.8 Метод Рунге.

Как видно из конечно-разностных соотношений для аппроксимаций производных, порядок их точности возрастает с увеличением числа узлов, используемых при аппроксимации. Однако при большом числе узлов эти соотношения становятся весьма громоздкими, что приводит к существенному возрастанию объёма вычислений. Усложняется также оценка точности получаемых результатов. Вместе с тем существует простой и эффективный способ уточнения решения при фиксированном числе узлов, используемых в аппроксимирующих конечно-разностных соотношениях. Это метод *Рунге*.

Пусть $F(x)$ — производная, которая подлежит аппроксимации; $f(x,h)$ — конечно-разностная аппроксимация этой производной на равномерной сетке с шагом h; R — погрешность (остаточный член) аппроксимации, главный член которой можно записать в виде $h^p\varphi(x)$, т. е.

$$R = h^p\varphi(x) + O(h^{p+1}).$$

(Для левой разности, например, $R = O(h) = h^1 \cdot \varphi(x) + O(h^2)$. Вспомним, что при $h \to 0$ справедливо $O(h) + O(h^2) = O(h)$.)

Тогда выражение для аппроксимации производной в общем случае можно представить в виде

$$F(x) = f(x,h) + h^p\varphi(x) + O(h^{p+1}). \tag{5.5}$$

(Для левой разности, например, $F(x) = \frac{f_i - f_{i-1}}{h} + h^1 \cdot \varphi(x) + O(h^2)$.)

Запишем это соотношение в той же точке x при другом шаге $h_1 = kh$. Получим

$$F(x) = f(x,kh) + (kh)^p\varphi(x) + O((kh)^{p+1}). \tag{5.6}$$

Приравнивая правые части равенств (5.5) и (5.6), находим выражение для главного члена погрешности аппроксимации производной:

$$h^p\varphi(x) = \frac{f(x,h) - f(x,kh)}{k^p - 1} + O(h^{p+1}).$$

Подставляя найденное выражение в равенство (5.5), получаем формулу *Рунге*

$$F(x) = f(x,h) + \frac{f(x,h) - f(x,kh)}{k^p - 1} + O(h^{p+1}). \tag{5.7}$$

Эта формула позволяет по результатам двух расчётов значений производной $f(x,h)$ и $f(x,kh)$ (с шагами h и kh) с порядком точности p найти её уточнённое значение с порядком точности $p+1$.

Пример. Рассмотрим формулу $y'(x) = \frac{y(x+h)-y(h)}{h} + O(h^1)$. Она имеет первый порядок погрешности. Получим с помощью метода Рунге формулу с бóльшим порядком погрешности. Убедимся вначале, что метод Рунге применим, то есть, что нашу формулу можно представить в виде (5.5). Вспомним разложение в ряд Тейлора:

$$y(x+h) = y(x) + y'(x)h + y''(\xi)\frac{h^2}{2!} = y(x) + y'(x)h + y''(x)\frac{h^2}{2!} + y'''(\eta)\frac{h^3}{3!},$$

где $\xi, \eta \in [x, x+h]$. Отсюда:

$$\underbrace{y'(x)}_{F(x)} = \underbrace{\frac{y(x+h)-y(x)}{h}}_{f(x,h)} + \underbrace{\frac{y''(\xi)}{2}h}_{O(h^1)} = \underbrace{\frac{y(x+h)-y(x)}{h}}_{f(x,h)} + \underbrace{\frac{y''(x)}{2}h}_{\varphi(x)h} + \underbrace{\frac{y'''(\eta)}{6}h^2}_{O(h^2)}.$$

Заключаем, что представление (5.5) верно. Следовательно, можно применять метод Рунге.

Подставим в (5.7) вместо $f(x)$ нашу формулу $\frac{y(x+h)-y(h)}{h}$ и положим $k = 2$:

$$F(x) = \frac{y(x+h)-y(h)}{h} + \frac{\frac{y(x+h)-y(h)}{h} - \frac{y(x+2h)-y(h)}{2h}}{2^1-1} + O(h^{1+1}) =$$
$$= \frac{1}{h}\left(-\frac{3}{2}y(x) + 2y(x+h) - \frac{1}{2}y(x+2h)\right) + O(h^2),$$

где $F(x) = y'(x)$. Можно переписать короче:

$$y'_i = \frac{1}{h}\left(-\frac{3}{2}y_i + 2y_{i+1} - \frac{1}{2}y_{i+2}\right) + O(h^2).$$

Полученная формула имеет уже второй порядок погрешности.

1.6 Лекция 6

Формулы для численного вычисления интегралов называются *квадратурными формулами* (от слова «квадрат», понимаемого в значении «площадь»).

Разобьём отрезок $[a, b]$, по которому интегрируют функцию на n частей с помощью точек $x_i, i = 0, 1, \ldots, n$, которые называют *узлами квадратурной формулы*. Тогда можно записать

$$\int_a^b f(x)\,dx = \sum_{k=1}^{n} \int_{x_{i-1}}^{x_i} f(x)\,dx.$$

1.6.1 Формула прямоугольников

Приблизим частичный интеграл площадью прямоугольника

$$\int_{x_{i-1}}^{x_i} f(x)\,dx \approx f(x_{i-1/2})h.$$

Это равносильно тому, что мы заменили функцию $f(x)$ на $[x_{i-1}, x_i]$ многочленом Лагранжа нулевой степени $P_0(x) = f(x_{i-1/2})$. Погрешность метода

$$\psi_i = \int_{x_{i-1}}^{x_i} f(x)\,dx - f(x_{i-1/2})h = \int_{x_{i-1}}^{x_i} [f(x) - f(x_{i-1/2})]\,dx.$$

С помощью формулы Тейлора

$$f(x) = f(x_{i-1/2}) + (x - x_{i-1/2})f'(x_{i-1/2}) + \frac{(x - x_{i-1/2})^2}{2} f''(\zeta_i),$$

где $\zeta_i = \zeta_i(x) \in [x, x_i]$. Откуда

$$\psi_i = \int_{x_{i-1}}^{x_i} \left[(x - x_{i-1/2})f'(x_{i-1/2}) + \frac{(x - x_{i-1/2})^2}{2} f''(\zeta_i) \right] dx =$$

$$= \int_{x_{i-1}}^{x_i} \frac{(x - x_{i-1/2})^2}{2} f''(\zeta_i)\,dx.$$

Оценка сверху

$$|\psi_i| \leqslant M_{2,i} \int\limits_{x_{i-1}}^{x_i} \frac{(x - x_{i-1/2})^2}{2}\, dx = \frac{h^3}{24} M_{2,i}, \text{ где } M_{2,i} = \max_{[x_{i-1},x_i]} |f''(x)|.$$

Суммируя частичные интегралы, получим *составную формулу прямоугольников*

$$\int\limits_a^b f(x)\, dx \approx (f_{1/2} + f_{3/2} + \ldots + f_{n-1/2})h$$

Погрешность этой формулы

$$\Psi = \int\limits_a^b f(x)\, dx - \sum_{i=1}^{n} f(x_{i-1/2})h = \sum_{i=1}^{n} \psi_i$$

$$|\Psi| \leqslant \sum_{i=1}^{n} |\psi_i| \leqslant \sum_{i=1}^{n} \frac{h^3}{24} M_{2,i} \leqslant \sum_{i=1}^{n} \frac{h^3}{24} M_2 = n\frac{h^3}{24} M_2 = \frac{h^2(b-a)}{24} M_2 = O(h^2),$$

где $M_2 = \max\limits_{[a,b]} |f''(x)|$ — максимум уже на всём отрезке $[a, b]$.

1.6.2 Формула трапеций

Приблизим частичный интеграл площадью трапеции

$$\int\limits_{x_{i-1}}^{x_i} f(x)\, dx \approx \frac{f(x_{i-1/2}) + f(x_i)}{2} h.$$

Это равносильно тому, что мы заменили функцию $f(x)$ на $[x_{i-1}, x_i]$ многочленом Лагранжа первой степени

$$P_{1,i}(x) = \frac{1}{h}[(x - x_{i-1})f(x_i) - (x - x_i)f(x_{i-1})].$$

Погрешность многочлена Лагранжа

$$R_{1,i}(x) = f(x) - P_{1,i}(x) = \frac{f''(\zeta_i(x))}{2!}(x - x_{i-1})(x - x_i).$$

Откуда погрешность формулы трапеций

$$\psi_i = \int\limits_{x_{i-1}}^{x_i} R_{1,i}(x)\, dx$$

Оценим погрешность сверху

$$|\psi_i| \leqslant \left| \int\limits_{x_{i-1}}^{x_i} R_{1,i}(x)\,dx \right| \leqslant \frac{M_{2,i}}{2} \left| \int\limits_{x_{i-1}}^{x_i} (x - x_{i-1})(x - x_i)\,dx \right| = \frac{M_{2,i}h^3}{12},$$

где $M_{2,i} = \max\limits_{[x_{i-1},x_i]} |f''(x)|$. Складывая частичные интегралы, получим *составную формулу трапеций*

$$\int\limits_{x_{i-1}}^{x_i} f(x)\,dx \approx \sum_{i=1}^{n} \frac{f(x_i) - f(x_{i-1})}{2} h = h\left(\frac{1}{2}f_0 + f_1 + \ldots + f_{n-1} + \frac{1}{2}f_n\right)$$

Погрешность составной формулы складывается из суммы частичных погрешностей $\Psi = \sum\limits_{i=1}^{n} \psi_i$. Справедлива оценка

$$|\Psi| \leqslant \sum_{i=1}^{n} |\psi_i| \leqslant \sum_{i=1}^{n} \frac{h^3}{12} M_{2,i} \leqslant \sum_{i=1}^{n} \frac{h^3}{12} M_2 = n\frac{h^3}{12}M_2 = \frac{h^2(b-a)}{12}M_2 = O(h^2),$$

где $M_2 = \max\limits_{[a,b]} |f''(x)|$.

1.6.3 Формула Симпсона

Приблизим функция $f(x)$ на частичном интервале многочленом Лагранжа 2-ой степени, который проходит через точки (x_{i-1}, f_{i-1}), $(x_{i-1/2}, f_{i-1/2})$ и (x_i, f_i)

$$f(x) \approx P_{2,i} = f_{i-1}\frac{L_2^{(i-1)}(x)}{L_2^{(i-1)}(x_{i-1})} + f_{i-1/2}\frac{L_2^{(i-1/2)}(x)}{L_2^{(i-1/2)}(x_{i-1/2})} + f_i\frac{L_2^{(i)}(x)}{L_2^{(i)}(x_i)} =$$

$$= f_{i-1}\frac{(x - x_{i-1/2})(x - x_i)}{h/2 \cdot h} - f_{i-1/2}\frac{(x - x_{i-1})(x - x_i)}{h/2 \cdot h/2} + \\ + f_i\frac{(x - x_{i-1})(x - x_{i-1/2})}{h \cdot h/2},$$

где $h = x_i - x_{i-1}$. Проведя интегрирование, получим

$$\int\limits_{x_{i-1}}^{x_i} f(x)\,dx \approx \int\limits_{x_{i-1}}^{x_i} P_{2,i}(x)\,dx = \frac{h}{6}(f_{i-1} + 4f_{i-1/2} + f_i).$$

Последнее выражение называется *формулой Симпсона*. На всём отрезке $[a,b]$ формула Симпсона имеет вид

$$\int_a^b f(x)\,dx \approx \sum_{i=1}^{n} \frac{h}{6}(f_{i-1}+4f_{i-1/2}+f_i) =$$
$$= \frac{h}{6}[f_0+f_n+2(f_1+f_2+\ldots+f_{n-1})+4(f_{1/2}+f_{3/2}+\ldots+f_{n-1/2})].$$

Чтобы не использовать дробных индексов, можно обозначить $x_i = a+\frac{1}{2}hi$, $i=0,1,\ldots,2n$, $hN=b-a$ и записать формулу Симпсона в виде

$$\int_a^b f(x)\,dx \approx \frac{b-a}{6n}[f_0+f_{2n}+2(f_2+f_4+\ldots+f_{2n-2})+4(f_1+f_3+\ldots+f_{2n-1})].$$

Для оценки погрешности потребуется вспомогательная

Лемма. *Формула Симпсона точна для любого многочлена 3-ей степени, т.е. имеет место точное равенство* $\int_{x_{i-1}}^{x_i} f(x)\,dx = \frac{h}{6}(f_{i-1}+4f_{i-1/2}+f_i)$, *если* $f(x)=a_0+a_1x+a_2x^2+a_3x^3$.

Доказательство получается непосредственной подстановкой.

Для оценки погрешности формулы Симпсона можно было бы воспользоватьс погрешностью многочлена Лагранжа

$$\psi_i = \int_{x_{i-1}}^{x_i} [f(x)-P_{2,i}(x)]\,dx = \int_{x_{i-1}}^{x_i} \frac{f'''(\zeta_i(x))}{3!}(x-x_{i-1})(x-x_{i-1/2})(x-x_i).$$

Откуда $|\psi_i| \sim h^4$, т.е. погрешность составной формулы Симпсона будет $|\Psi| \leqslant \sum\limits_{i=1}^{n} |\psi_i| \sim h^3$. Но качество численной формулы измеряется порядком погрешности. Покажем, как можно повысить в оценке порядок погрешности.

Воспользуемся интерполяционным многочленом Эрмита. Построим многочлен 3-ей степени $H_3(x)$ такой, что

$$H_3(x_{i-1}) = f(x_{i-1}), \qquad H_3(x_{i-1/2}) = f(x_{i-1/2}),$$
$$H_3(x_i) = f(x_i), \qquad H_3'(x_{i-1/2}) = f'(x_{i-1/2}).$$

Привлекая последнюю лемму, получим что

$$\int\limits_{x_{i-1}}^{x_i} H_3(x)\,dx \overset{\text{лемма}}{=} \frac{h}{6}[H_3(x_{i-1})+4H_3(x_{i-1/2})+H_3(x_i)] = \frac{h}{6}[f_{i-1}+4f_{i-1/2}+f_i].$$

Погрешность на i-ом частичном интервале

$$\psi_i = \int\limits_{x_{i-1}}^{x_i} f(x)\,dx - \frac{h}{6}(f_{i-1}+4f_{i-1/2}+f_i) =$$

$$= \int\limits_{x_{i-1}}^{x_i} f(x)\,dx - \frac{h}{6}[H_3(x_{i-1})+4H_3(x_{i-1/2})+H_3(x_i)] =$$

$$= \int\limits_{x_{i-1}}^{x_i} [f(x)-H_3(x)]\,dx = \int\limits_{x_{i-1}}^{x_i} R_{3,i}(x)\,dx.$$

Как известно, для многочлена Эрмита погрешность равна

$$R_{3,i}(x) = \frac{f^{\mathrm{IV}}(\zeta_i(x))}{4!}(x-x_{i-1})(x-x_{i-1/2})^2(x-x_i).$$

Поэтому

$$|\psi_i| \leqslant \frac{M_{4,i}}{24}\left|\int\limits_{x_{i-1}}^{x_i}(x-x_{i-1})(x-x_{i-1/2})^2(x-x_i)\,dx\right| = \frac{h^5}{2880}M_{4,i},$$

где $M_{4,i} = \max\limits_{[x_{i-1},x_i]}|f^{\mathrm{IV}}(x)|$. Для составной формулы Симпсона погрешность

$$|\Psi| \leqslant \frac{h^4(b-a)}{2880}M_4, \qquad \text{где } M_4 = \max_{[a,b]}|f^{\mathrm{IV}}(x)|.$$

Замечание. *Может показаться, что формула Симпсона точнее, формулы прямоугольников или формулы трапеций. В общем случае это не так. Например, погрешности* $|\Psi_{\text{прям}}| \leqslant \frac{h^2(b-a)}{24}M_2$ *и* $|\Psi_{\text{симп}}| \leqslant \frac{h^4(b-a)}{2880}M_4$ *зависят не только от шага* h*, но и от соответствующих производных. С уверенностью можно утверждать только, что скорость уменьшения погрешности при уменьшении шага* h *у формулы Симпсона будет больше, чем у формулы прямоугольников или формулы трапеций. Например,*

Шаг	*Формула с погрешн.* $O(h^2)$	*Формула с погрешн.* $O(h^4)$
h	*погрешность* ε	*погрешность* ε
$h/2$	*погрешность* $\varepsilon/4$	*погрешность* $\varepsilon/16$
$h/4$	*погрешность* $\varepsilon/16$	*погрешность* $\varepsilon/256$

Оценки для погрешностей $\Psi_{\text{прям}}$, $\Psi_{\text{трап}}$ и $\Psi_{\text{симп}}$ были получены в предположении существования непрерывных производных соответствующего порядка. Например, если у функции нет 4-ой непрерывной на $[a, b]$ производной, то порядок погрешности формулы Симпсона будет меньше, чем 4.

1.7 Лекция 7

1.7.1 Метод Рунге

Квадратурные формулы имеют погрешность вида $O(h^p)$. Если удастся выделить главный член в погрешности, то можно с помощью метода Рунге получить формулу с большим порядком погрешности. Для примера рассмотрим формулу трапеций. Запишем частичную погрешность более подробно, чем раньше[2]

$$\psi_i = \int\limits_{x_{i-1}}^{x_i} f(x)\,dx - h\frac{f_{i-1}+f_i}{2} \overset{\text{погр. мн. Лагр.}}{=} \int\limits_{x_{i-1}}^{x_i} \frac{f''(\xi(x))}{2!}(x-x_{i-1})(x-x_i)\,dx =$$

$$\overset{\text{ф-ла сред. знач.}}{=} \frac{f''(\zeta_i)}{2!}\int\limits_{x_{i-1}}^{x_i}(x-x_{i-1})(x-x_i)\,dx = -\frac{f''(\zeta_i)h^3}{12}, \quad \text{где } \zeta_i \in [x_{i-1}, x_i].$$

Общая погрешность на интервале $[a,b]$ будет

$$\Psi = \sum_{i=1}^{n}\psi_i = -\frac{h^2}{12}\sum_{i=1}^{n} f''(\zeta_i)h \overset{\text{при } h\to 0}{=}$$

$$= -\frac{h^2}{12}\left(\int\limits_a^b f''(x)\,dx + O(h)\right) \overset{\text{ф-ла Н.-Л.}}{=} -\frac{h^2}{12}(f'(b)-f'(a)) + O(h^3)$$

Главный член погрешности получен. Обозначим

$$S_{\text{тр}}(h) = h\left(\frac{1}{2}f_0 + f_1 + f_2 + \ldots + f_{n-1} + \frac{1}{2}f_n\right).$$

$$\int\limits_a^b f(x)\,dx \begin{array}{l} \nearrow S_{\text{тр}}(h) + ch^2 + O(h^3) \\ \searrow S_{\text{тр}}(h/2) + c(h/2)^2 + O(h^3) \end{array}$$

[2]Здесь потребуется известная из математического анализа

Теорема (формула среднего значения)**.** *Пусть функция $f(x)$ непрерывна на $[a,b]$, функция $g(x)$ интегрируема на $[a,b]$ и $g(x) \geqslant 0$ (или $g(x) \leqslant 0$) на всём $[a,b]$, тогда существует такое $\xi \in [a,b]$, что*

$$\int\limits_a^b f(x)g(x)\,dx = f(\xi)\int\limits_a^b g(x)\,dx.$$

где $c = -\frac{f'(b)-f'(a)}{12}$. Выразим неизвестное слагаемое

$$ch^2 = \frac{S_{\text{тр}}(h/2) - S_{\text{тр}}(h)}{1 - 1/4} + O(h^3).$$

Отсюда

$$\int_a^b f(x)\,dx = S_{\text{тр}}(h) + \frac{4}{3}[S_{\text{тр}}(h/2) - S_{\text{тр}}(h)] + O(h^3) =$$

$$= \frac{4S_{\text{тр}}(h/2) - S_{\text{тр}}(h)}{3} + O(h^3).$$

Заменим теперь $S_{\text{тр}}$

$$\int_a^b f(x)\,dx = \frac{1}{3}\left[4\frac{h}{2}\left(\frac{1}{2}f_0 + f_{1/2} + f_1 + \ldots + f_{n-1/2} + \frac{1}{2}f_n\right) + \right.$$

$$\left. + h\left(\frac{1}{2}f_0 + f_1 + f_2 + \ldots + f_{n-1} + \frac{1}{2}f_n\right)\right] + O(h^3) =$$

$$= \frac{h}{6}[f_0 + f_n + 2(f_1 + f_2 + \ldots + f_{n-1}) + 4(f_{1/2} + f_{3/2} + \ldots + f_{n-1/2})] + O(h^3).$$

В итоге получили больший порядок погрешности ошибки, а сама квадратурная формула совпадает с формулой Симпсона.

1.7.2 Оценка погрешности

Пусть требуется получить приближённое значение интеграла с точностью ε. Возникает вопрос, какой следует выбрать шаг h. Можно попробовать вычислить оценку погрешности для используемой формулы. В этом случае придётся иметь дело с производными. Например, вычисляя по формуле Симпсона, потребуется оценить $f^{\text{IV}}(x)$. В общем случае это сделать непросто. На практике часто используют более удобный метод Рунге.

Пусть $S(h)$ обозначает квадратурную формулу (например, трапеций или Симпсона) с шагом h, которая используется в наших расчётах на компьютере. Справедливо представление

$$I = \int_a^b f(x)\,dx = S(h) + O(h^p) = S(h) + ch^p + O(h^{p+1}),$$

где p — порядок погрешности формулы $S(h)$, а c — константа. Проведём последовательно расчёт на компьютере с шагом h, а затем $h/2$. Тогда интеграл можно приближённо, без учёта $O(h^{p+1})$ выразить двумя способами

$$\int_a^b f(x)\,dx \begin{array}{l} \curvearrowright S_{\text{тр}}(h) + ch^p \\ \curvearrowright S_{\text{тр}}(h/2) + c(h/2)^p \end{array}$$

Наша цель — это проверка условия $|I - S(h/2)| \leqslant \varepsilon$. Имеем

$$|I - S(h/2)| \approx |c(h/2)^p| \approx \frac{|S(h) - S(h/2)|}{2^p - 1} \leqslant \varepsilon.$$

Приведём таблицу для рассмотренных ранее квадратурных формул.

ф. прямоугольников	ф. трапеций	ф. Симпсона
$\|I - S(h/2)\| \approx \frac{\|S(h)-S(h/2)\|}{3} \leqslant \varepsilon$		$\|I - S(h/2)\| \approx \frac{\|S(h)-S(h/2)\|}{15} \leqslant \varepsilon$

Проверка погрешности сводится к проверке неравенства для расчётов с шагами h и $h/2$. Если неравенство не выполняется, уменьшаем шаг ещё в два раза и подставляем $S(h/2)$ и $S(h/4)$ и т.д.

Метод можно использовать для выбора шага в зависимости от скорости роста функции $f(x)$ (скорость роста определяется величиной $f'(x)$). Когда скорость роста не велика, можно использовать широкий шаг (рис. 7.1). На участках резкого изменения функции шаг лучше выбрать частый. В данном подходе в отличии от всюду одинакового шага можно сэкономить на количестве использованных узлов.

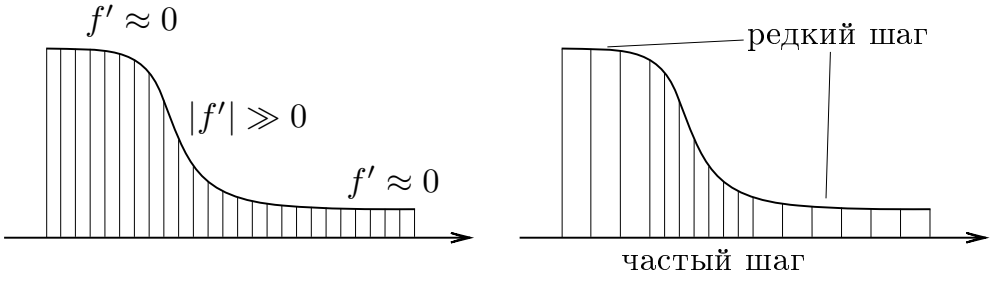


Рис. 7.1: а) постоянный шаг на всём отрезке $[a, b]$; б) выбор шага в зависимости от скорости роста функции (от величины $f'(x)$)

На практике мы разбиваем весь отрезок $[a, b]$ на p больших частей (p — несколько единиц)

$$[a, b] = [a = c_0, c_1] \cup [c_1, c_2] \cup \ldots \cup [c_{p-1}, c_p = b].$$

Если задана погрешность для ответа ε, то на каждой части $[c_{i-1}, c_i]$, $i = 1, 2, \ldots, p$ зададим погрешность ε/p. Далее с каждым отрезком $[c_{i-1}, c_i]$ работаем отдельно, выбирая свой шаг по методу Рунге.

1.7.3 Формулы Ньютона–Котеса

Приблизим функцию многочленом n-ой степени $f(x) \approx P_n(x)$. Тогда интеграл от $f(x)$ можно приблизить интегралом от многочлена

$$\int_a^b f(x)\,dx \approx \int_a^b P_n(x)\,dx.$$

Возникающая при этом погрешность равна

$$\Psi = \int_a^b [f(x) - P_n(x)]\,dx \overset{\text{погр. мн. Лагр.}}{=} \int_a^b \frac{f^{(n+1)}(\xi(x))}{(n+1)!}\omega_{n+1}(x)\,dx.$$

Оценивая сверху, получим

$$|\Psi| \leqslant \frac{M_{n+1}}{(n+1)!}\int_a^b |\omega_{n+1}(x)|\,dx, \quad \text{где } M_{n+1} = \max_{[a,b]} |f^{(n+1)}(x)|.$$

Рассмотрим подробнее интеграл от многочлена

$$\int_a^b P_n(x)\,dx = \int_a^b \left[\sum_{k=0}^n f_k \frac{L_n^{(k)}(x)}{L_n^{(k)}(x_k)}\right] dx = \sum_{k=0}^n f_k \left[\int_a^b \frac{L_n^{(k)}(x)}{L_n^{(k)}(x_k)}\,dx\right] = \sum_{k=0}^n f_k c_k,$$

где $c_i = \int_a^b L_n^{(k)}(x)\,dx / L_n^{(k)}(x_k)$. Видно, что коэффициенты c_i являются константами, значения, которых не зависят от интегрируемой функции, но определяются только узлами $x_0 = a, x_1, x_2, \ldots, x_n = b$. Следовательно, для заданных интервала $[a, b]$ и шага h можно один раз рассчитать c_i, $i = 0, 1, \ldots, n$ и использовать дальше для численного вычисления любых функций по формуле

$$\int_a^b f(x)\,dx \approx \sum_{k=0}^n f_k c_k. \tag{7.1}$$

Последняя формула в случае постоянного шага $h = x_k - x_{k-1} = \text{const}$ носит название *формулы Ньютона-Ко́теса*[3].

Исследуем теперь возможность поставить в (7.1) знак $=$ вместо $\approx$. Пусть $f(x) = a_0 + a_1 x + \ldots + a_m x^m$, $a_m \neq 0$ — многочлен степени m. Приблизим $f(x)$ многочленом Лагранжа $P_n(x)$ степени n. Ясно, что $f(x) = P_n(x)$, если $m \leqslant n$ и, следовательно, $\int_a^b f(x)\,dx = \int_a^b P_n(x)\,dx = \sum_{k=0}^n f_k c_k$.

Оказывается, в случае чётного n формула (7.1) будет точна (равенство будет не приближённым, а точным) даже для $m = n + 1$.

Лемма. *Если n — чётное, то для коэффициентов формулы Ньютона–Котеса справедливо $c_k = c_{n-k}$, где $k = 0, 1, \ldots, n/2$.*

Доказательство. □

Теорема. *Если n — чётное и $P_{n+1}(x)$ — многочлен степени $n + 1$, то справедливо равенство*

$$\int_a^b P_{n+1}(x)\,dx = \sum_{k=0}^n f_k c_k.$$

В заключении отметим, что формулы Ньютона–Котеса порядка $\geqslant 10$ или $n = 8$ не применяют из-за того, что коэффициенты c_k, $k = 0, 1, \ldots n$ имеют разный знак (это важно для устойчивости, см. далее). При вычислении интегралов на длинных интервалах $[a, b]$, сам интервал $[a, b]$ делится на несколько частей (частичных отрезков). На каждом частичном отрезке строится формула Ньютона–Котеса невысокой степени. В итоге на всём отрезке $[a, b]$ получается *составная формула Ньютона–Котеса*. Примеры составных формул уже встречались при рассмотрении методов прямоугольников, трапеций и Симпсона.

[3] Ро́джер Ко́тес (Roger Cotes) — английский математик (1682–1716).

n	Формула Ньютона–Котеса (несоставные)
2	$\frac{h}{3}(f_0 + 4f_1 + f_2)$
3	$\frac{3h}{8}(f_0 + 3f_1 + 3f_2 + f_3)$
4	$\frac{2h}{45}(7f_0 + 32f_1 + 12f_2 + 32f_3 + 7f_4)$
5	$\frac{5h}{288}(19f_0 + 75f_1 + 50f_2 + 50f_3 + 75f_4 + 19f_5)$
6	$\frac{h}{140}(41f_0 + 216f_1 + 27f_2 + 272f_3 + 27f_4 + 216f_5 + 41f_6)$

1.8 Лекция 8

Лемма 8.1. *Если в формуле Ньютона–Котеса n — чётно, тогда $c_k = c_{n-k}$.*

Доказательство. Имеем

$$c_k = \frac{\int\limits_{x_0}^{x_n} L_n^{(k)}(x)\,dx}{L_n^{(k)}(x_k)}, \quad c_{n-k} = \frac{\int\limits_{x_0}^{x_n} L_n^{(n-k)}(x)\,dx}{L_n^{(n-k)}(x_{n-k})}. \tag{8.1}$$

По определению формулы Ньютона–Котеса $x_i - x_{i-1} = h = \text{const}$. Отсюда следует, что для любых $i, j \in \{0, 1, \ldots, n\}$ справедливо $x_i - x_j = (i-j)h$.

Для знаменателей (8.1) получаем, что

$$L_n^{(k)}(x_k) = \\ = (k-0)h \cdot (k-1)h \cdot \ldots \cdot 2h \cdot 1h \cdot \underbrace{(-1)h \cdot (-2)h \cdot \ldots \cdot (k-n)h}_{n-k \text{ отриц. множ.}} = \\ = k!\,(n-k)!\,h^n\,(-1)^{n-k},$$

$$L_n^{(n-k)}(x_{n-k}) = \\ = (n-k-0)h \cdot (n-k-1)h \cdot \ldots \cdot 2h \cdot 1h \cdot \underbrace{(-1)h \cdot (-2)h \cdot \ldots \cdot (n-k-n)h}_{k \text{ отриц. множ.}} = \\ = (n-k)!\,k!\,h^n\,(-1)^k.$$

Так как n — чётное, то числа k и $n-k$ — чётные или нечётные одновременно. Следовательно, $L_n^{(k)}(x_k) = L_n^{(n-k)}(x_{n-k})$.

Перейдем теперь к числителям (8.1). Заметим, что $L_n^{(k)}(x) = \omega_{n+1}(x)/(x - x_k)$. Так как n — чётно, то $n_{n/2}$ является серединой отрезка $[x_0, x_n]$. Остальные узлы x_k, $k = 0, 1, \ldots, n$ расположены симметрично относительно $n_{n/2}$. Обозначим $t_k = x_{n/2} - x_k = x_{n-k} - x_{n/2}$, тогда

$$x - x_k = (x - x_{n/2}) + (x_{n/2} - x_k) = x - x_{n/2} + t_k,$$

$$x - x_{n-k} = (x - x_{n/2}) + (x_{n/2} - x_{n-k}) = x - x_{n/2} - t_k.$$

Рассмотрим разность числителей из (8.1)

$$\int_{x_0}^{x_n} L_n^{(k)}(x)\,dx - \int_{x_0}^{x_n} L_n^{(n-k)}(x)\,dx = \int_{x_0}^{x_n} \left(\frac{\omega_{n+1}(x)}{x-x_k} - \frac{\omega_{n+1}(x)}{x-x_{n-k}}\right) dx =$$

$$= \int_{x_0}^{x_n} \omega_{n+1}(x) \left(\frac{1}{x-x_{n/2}+t_k} - \frac{1}{x-x_{n/2}-t_k}\right) dx = -2t_k \int_{x_0}^{x_n} \frac{\omega_{n+1}(x)\,dx}{(x-x_{n/2})^2 - t^2}.$$

Функция $\omega_{n+1}(x)$ — нечётна относительно $x = x_{n/2}$, функция $(x-x_{n/2})^2 - t^2$ — чётна относительно $x = x_{n/2}$. Следовательно, всё подинтегральное выражение является нечётной функций относительно $x = x_{n/2}$. Так как $x_{n/2}$ есть середина отрезка $[x_0, x_n]$, по которому происходит интегрирование, то интеграл равен нулю.

Мы показали, что в (8.1) числители и знаменатели равны. Следовательно, $c_k = c_{n-k}$. □

Теорема 8.2. *Формула Ньютона–Котеса, где n — чётное число, точна для любого многочлена степени $n+1$.*

Доказательство. Пусть $f(x) = a_{n+1}x^{n+1} + a_n x^n + \ldots + a_1 x + a_0$. Формула Ньютона-Котеса имеет вид

$$\int_a^b f(x)\,dx \approx c_0 f(x_0) + c_1 f(x_1) + \ldots + c_n f(x_n).$$

Погрешность формулы будет равна

$$\varepsilon = \int_a^b f(x)\,dx - (c_0 f(x_0) + c_1 f(x_1) + \ldots + c_n f(x_n)) = \int_a^b f(x)\,dx - \int_a^b P_n(x)\,dx,$$

где $P_n(x)$ — интерполяционный многочлен Лагранжа степени n.

$$\varepsilon = \int_a^b \frac{f^{(n+1)}(x(\xi))}{(n+1)!}\omega_{n+1}(x)\,dx.$$

Очевидно $f^{(n+1)}(x) = a_{n+1} \cdot (n+1)!$ Поэтому

$$\varepsilon = a_{n+1} \int_a^b \omega_{n+1}(x)\,dx = a_{n+1} \int_a^b (x-x_0)(x-x_1)\ldots(x-x_n)\,dx.$$

Сделаем замену $t = x - x_{n/2}$.

$$\varepsilon = a_{n+1} \int\limits_{a-x_{n/2}}^{b-x_{n/2}} (t + x_{n/2} - x_0)(t + x_{n/2} - x_1) \dots (t + x_{n/2} - x_n)\, dx.$$

Вспомним, что $a = x_0$, $b = x_n$, середина отрезка $[a, b]$ — это $\frac{a+b}{2} = x_{n/2}$. Кроме этого $x_i - x_{i-1} = h$. Отсюда

$$b - x_{n/2} = -\frac{a+b}{2},$$

$$a - x_{n/2} = \frac{a+b}{2},$$

$$x_{n/2} - x_0 = \frac{n}{2}h,$$

$$x_{n/2} - x_1 = \left(\frac{n}{2} - 1\right) h,$$

$$\vdots$$

$$x_{n/2} - x_n = \left(\frac{n}{2} - n\right) h = -\frac{n}{2}h.$$

Вернёмся к погрешности

$$\varepsilon = a_{n+1} \int\limits_{-(a+b)/2}^{(a+b)/2} \left(t - \frac{n}{2}h\right) \left(t - \left(\frac{n}{2} - 1\right) h\right) \cdots$$

$$\cdots (t-h)t(t+h) \cdots$$

$$\cdots \left(t + \left(\frac{n}{2} - 1\right) h\right) \left(t + \frac{n}{2}h\right)\, dx.$$

Пределы интегрирования симметричны, подинтегральная функция нечетная. Интеграл равен нулю и погрешность также равна нулю.

□

1.8.1 Устойчивость квадратурных формул к погрешностям входных данных

Изучим влияние погрешности входных данных на результат численного интегрирования. Рассмотрим в общем виде квадратурную формулу

Ньютона–Котеса (неважно простую или составную)

$$\int_a^b f(x)\,dx \approx \sum_{k=0}^{n} c_k f(x_k) = I_n. \tag{8.2}$$

Заметим сразу, что для функции $f(x) \equiv 1$ формула I_n точна, т.е.

$$\int_a^b 1\,dx = \sum_{k=0}^{n} c_k \cdot 1 \text{ и, следовательно, } \sum_{k=0}^{n} c_k = b - a.$$

Пусть из-за округлений или неточных измерений входные данные содержат погрешность. Т.е. вместо точных значений $f(x_k)$, $k = 0, 1, \ldots, n$ мы располагаем $\widetilde{f}(x_k) = f(x_k) + \delta_k$, где δ_k — погрешность в точке x_k. Вместо (8.2) получим

$$\widetilde{I}_n = \sum_{k=0}^{n} c_k \widetilde{f}(x_k) = \sum_{k=0}^{n} c_k f(x_k) + \sum_{k=0}^{n} c_k \delta_k = I_n + \delta I_n, \text{ где } \delta I_n = \sum_{k=0}^{n} c_k \delta_k.$$

Возможны два случая: (а) все $c_k > 0$, $k = 0, 1, \ldots, n$ и (б) не все c_k одного знака.

В случае (а) имеем оценку

$$|\delta I_n| \leqslant \sum_{k=0}^{n} |c_k||\delta_k| = \sum_{k=0}^{n} c_k |\delta_k| \leqslant (\max_k |\delta_k|) \sum_{k=0}^{n} c_k = (b-a) \max_k |\delta_k|,$$

которая означает, что погрешность δI_n не зависит от количества узлов n (или, что то же самое, от величины шага h). δI_n пропорциональна наибольшей из погрешностей входных данных $\max\limits_k |\delta_k|$. Случай (а) *устойчив*.

В случае (б) $\sum\limits_{k=0}^{n} |c_k| \geqslant \sum\limits_{k=0}^{n} c_k$. Сумма $\sum\limits_{k=0}^{n} c_k = b - a$ равномерно ограничена по n, так как её величина не зависит от количества разбиений отрезка $[a, b]$. Напротив, сумма $\sum\limits_{k=0}^{n} |c_k|$ может не оказаться равномерно ограниченной по n и с ростом n будет неограниченно возрастать. Следовательно, ошибка

$$|\delta I_n| \leqslant \sum_{k=0}^{n} |c_k||\delta_k|$$

уже не будет пропорциональна $\max\limits_k |\delta_k|$ (будет намного больше). Случай (б) *неустойчив*.

Подводя итог, скажем, что для устойчивости квадратурных формул необходима положительность коэффициентов c_k. Следовательно, формулы Ньютона–Котеса (простые или составные) должны быть основаны на многочленах Лагранжа степени $n \leqslant 9$ и $n \neq 8$. При $n = 8$ или $n \geqslant 10$ коэффициенты c_k меняют знак.

1.8.2 Приёмы вычисления несобственных интегралов

Будем рассматривать сходящиеся интегралы двух типов

1. $\int\limits_a^b f(x)\,dx$, причём $f(x) \to \infty$ при $x \to a$.
2. $\int\limits_a^\infty f(x)\,dx$.

Второй интеграл можно свести к первому заменой переменной $t = \frac{1}{x}$, $dx = -\frac{dt}{t^2}$, $\int\limits_0^{1/a} \frac{f(1/t)}{t^2}\,dt$.

Рассмотрим интегралы первого типа. Непосредственное применение формул трапеций или Симпсона невозможно (так как в узле интегрирования $x = a$ функция $f(x)$ неопределена). Использование формулы прямоугольников возможно, но оценка точности теряет смысл, так как $f'(0)$ неопределена.

Пример 1. Продемонстрируем приёмы, позволяющие получить надёжные результаты в подобных случаях, на примере интеграла

$$I = \int\limits_0^1 \frac{\cos x}{\sqrt{x}}\,dx.$$

а) Подходящая замена переменной

$$x = t^2, \quad dx = 2tdt = 2\sqrt{x}dt, \quad I = 2\int\limits_0^1 \cos(t^2)\,dt.$$

Далее можно проводить вычисления с требуемой точностью по любой квадратурной формуле.

б) Интегрирование по частям

$$I = \int_0^1 \frac{\cos x}{\sqrt{x}}\,dx = 2\sqrt{x}\cos x|_0^1 + 2\int_0^1 \sqrt{x}\sin x\,dx.$$

Последний интеграл можно вычислить численно, но оценка погрешности равна $O(h)$, т.к. $(\sqrt{x}\sin x)'|_{x=0}$ не существует. Если ещё раз проинтегрировать по частям, то под знаком интеграла окажется функция $f(x) \in C^2[0,1]$ (дважды непрерывно дифференцируемая на отрезке $[0,1]$). В этом случае ошибка будет $O(h^2)$.

в) Разбиение на два интеграла

$$I = I_1 + I_2, \quad I_1 = \int_0^\delta \frac{\cos x}{\sqrt{x}}\,dx, \quad I_2 = \int_\delta^1 \frac{\cos x}{\sqrt{x}}\,dx.$$

Второй интеграл I_2 не содержит особенности и может быть вычислен по любой квадратурной формуле с точностью $\varepsilon/2$. Первый интеграл I_1 вычисляется аналитически после замены $\cos x$ соответствующим рядом Тейлора

$$I_1 = \int_0^\delta \frac{1 - \frac{x^2}{2!} + \frac{x^4}{4!} + \ldots + (-1)^m \frac{x^{2m}}{(2m)!} + \ldots}{\sqrt{x}}\,dx = 2\sqrt{\delta} - \frac{1}{2!}\frac{2}{5}\delta^{5/2} + \frac{1}{4!}\frac{2}{9}\delta^{9/2} + \\ + \ldots + (-1)^m \frac{1}{2m!}\frac{1}{2m+1/2}\delta^{2m+1/2} + \ldots$$

Нам нужно получить значение I_1 с точность $\varepsilon/2$. Ряд в разложении I_1 удовлетворяет условиям сходимости признака Лейбница[4], поэтому, если

[4]

Теорема (признак Лейбница)**.** *Если члены знакочередующегося ряда*

$$S = p_1 - p_2 + p_3 - \ldots + (-1)^{k-1}p_k + \ldots, \text{ где все } p_k \geqslant 0,$$

будучи взяты по модулю, образуют невозрастающую бесконечно малую последовательность, то этот ряд сходится.

Следствие. *Если ряд удовлетворяет признаку сходимости Лейбница, тогда для его частичных сумм справедливо неравенство*

$$|S_n - S| \leqslant p_n.$$

отбросить слагаемые с номерами большими m, мы допустим погрешность не более, чем модуль последнего оставшегося слагаемого

$$\frac{1}{2m!}\frac{1}{2m+1/2}\delta^{2m+1/2} \leqslant \frac{\varepsilon}{2}.$$

Считаем, что точность ε фиксирована. Чтобы выполнялось последнее неравенство, мы будем варьировать m и δ.

Если выбрать очень малое δ ($\ll 0{,}1$) потребуется совсем немного слагаемых m, но от такого δ пострадает погрешность интеграла I_2, куда входит величина M_2 (для формулы прямоугольников или трапеций) или M_4 (для формулы Симпсона). В самом деле, M_2 содержит слагаемое $\sim \delta^{-3/2}$, а M_4 — слагаемое $\sim \delta^{-5/2}$. При малых δ это будут большие числа.

Если выбрать большое δ ($\gg 0{,}1$), погрешность у I_2 будет нормальная, но нужно много слагаемых m. Компромиссом можно считать «среднее» $\delta = 0{,}1$.

В итоге оба интеграла I_1 и I_2 имеют погрешность не более $\varepsilon/2$, а общая погрешность для $I = I_1 + I_2$ не превзойдёт $\varepsilon/2 + \varepsilon/2 = \varepsilon$.

Пример 2. Вычислим интеграл второго типа $I = \int\limits_0^\infty e^{-x^2}\,dx$. Данный интеграл можно свести к интегралу первого типа, но мы поступим иначе.

$$I = I_1 + I_2, \quad I_1 = \int\limits_0^A e^{-x^2}\,dx, \quad I_2 = \int\limits_A^\infty e^{-x^2}\,dx.$$

Выберем A таким образом, чтобы величиной I_2 можно было пренебречь, т.е. $|I_2| \leqslant \varepsilon/2$. Например, при $A > 1$

$$\int\limits_A^\infty e^{-x^2}\,dx \leqslant \int\limits_A^\infty xe^{-x^2}\,dx = \frac{1}{2}e^{-A^2}.$$

Потребуем, чтобы $\frac{1}{2}e^{-A^2} \leqslant \frac{\varepsilon}{2}$, откуда $A \geqslant \sqrt{|\ln\varepsilon|}$.

Делее вычислим I_1 стандартными методами с точностью $\varepsilon/2$.

Мы получили значение интеграла I_1 с погрешностью $\varepsilon/2$, а I_2 не превосходит $\varepsilon/2$. Следовательно, общая погрешность $I = I_1 + I_2$ будет не более $\varepsilon/2 + \varepsilon/2 = \varepsilon$.

1.9 Лекция 9

1.9.1 Элементы линейной алгебры

Норма вектора — это отображение из $\mathbb{R}^n$ в $\mathbb{R}$, обозначаемое $\|\mathbf{x}\|$ и удовлетворяющее свойствам:

1) $\|\mathbf{x}\| \geqslant 0$, $\|\mathbf{x}\| = 0 \Leftrightarrow \mathbf{x} = \mathbf{0}$,

2) $\|\alpha\mathbf{x}\| = |\alpha| \cdot \|\mathbf{x}\|$, α — скаляр,

3) $\|\mathbf{x} + \mathbf{y}\| \leqslant \|\mathbf{x}\| + \|\mathbf{y}\|$.

Примеры:

1. $\|\mathbf{x}\|_1 = \sum\limits_i |x_i|$,

2. $\|\mathbf{x}\|_2 = \sqrt{\sum\limits_i x_i^2}$ — евклидова норма,

3. $\|\mathbf{x}\|_\infty = \|\mathbf{x}\|_c = \max\limits_i |x_i|$ — равномерная норма.[5]

Векторное пространство с введённой в нём нормой называют *нормированным*. Одновременно оно является метрическим, так как норма определяет метрику — расстояние между элементами пространства:

$$\rho(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|.$$

Норма квадратной матрицы A — это отображение из $\mathbb{R}^{n\times n}$ в $\mathbb{R}$, обозначаемое $\|A\|$ и удовлетворяющее свойствам:

1) $\|A\| \geqslant 0$, $\|A\| = 0 \Leftrightarrow A = 0$ (матрица размера $n \times n$ из нулей),

2) $\|\alpha A\| = |\alpha| \cdot \|A\|$, α — скаляр,

3) $\|A + B\| \leqslant \|A\| + \|B\|$,

4) $\|AB\| \leqslant \|A\| \cdot \|B\|$.

Норма матрицы A *согласована* с нормой вектора $\mathbf{x}$, если

$$\|A\mathbf{x}\| \leqslant \|A\| \cdot \|\mathbf{x}\|.$$

[5] Все три нормы — это частные случаи *Гёльдеровой нормы* $\|\mathbf{x}\|_p = (\sum |x_i|^p)^{1/p}$ для $p = 1, 2, \infty$.

Норма матрицы A называется *подчинённой* норме вектора $\mathbf{x}$, если $\|A\|$ вводится следующим образом:

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

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

$$\frac{\|A\mathbf{x}\|}{\|\mathbf{x}\|} \leqslant \sup_{\mathbf{x}\neq\mathbf{0}} \frac{\|A\mathbf{x}\|}{\|\mathbf{x}\|} = \|A\|,$$

отсюда

$$\|A\mathbf{x}\| \leqslant \|A\| \cdot \|\mathbf{x}\|.$$

В дальнейшем интерес будут представлять согласованные нормы. Но таких норм может оказаться много. Чтобы избежать неоднозначности, выбирают единственную подчинённую норму, которая в то же время является согласованной.

Вывод формул для вычисления подчинённым матричных норм $\|\cdot\|_1$, $\|\cdot\|_2$ и $\|\cdot\|_\infty$ приведён в задаче 6.6 в главе «Практические занятия».

Следует отметить, что в конечномерном линейном пространстве все нормы эквивалентны в том смысле, что, если имеет место $\|\mathbf{x}_n\|_\alpha \underset{n\to\infty}{\longrightarrow} 0$ для бесконечной последовательности $\{\mathbf{x}_n\}$ в некоторой норме α, то в любой другой норме β также $\|\mathbf{x}_n\|_\beta \underset{n\to\infty}{\longrightarrow} 0$.

Пусть для данной матрицы A найдётся такой ненулевой вектор $\mathbf{x}$, что $A\mathbf{x} = \lambda\mathbf{x}$, где $\lambda \in \mathbb{R}$. Тогда $\mathbf{x}$ называется *собственным вектором*, а λ — *собственным значением*.

Лемма 9.1. *Пусть λ — собственное значение матрицы A и $\det A \neq 0$, тогда $1/\lambda$ — собственное значение матрицы A^{-1}.*

Доказательство. Так как $\det A \neq 0$, то матрица A^{-1} существует. Умножим равенство $A\mathbf{x} = \lambda\mathbf{x}$ слева на A^{-1}

$$A^{-1}A\mathbf{x} = A^{-1}\lambda\mathbf{x} \qquad \text{откуда} \qquad A^{-1}\mathbf{x} = \frac{1}{\lambda}\mathbf{x}.$$

□

Матрица A называется *положительно определённой* ($A > 0$) (неотрицательно определённой, $A \geqslant 0$), если $(A\mathbf{x}, \mathbf{x}) > 0$ ($(A\mathbf{x}, \mathbf{x}) \geqslant 0$) для любых $\mathbf{x} \neq \mathbf{0}$.

Пусть $A > 0$ и $\mathbf{x}$ — собственный вектор матрицы A, тогда $A\mathbf{x} = \lambda\mathbf{x}$ и

$$(A\mathbf{x}, \mathbf{x}) = (\lambda\mathbf{x}, \mathbf{x}) = \lambda(\mathbf{x}, \mathbf{x}).$$

Из $(A\mathbf{x}, \mathbf{x}) > 0$ и $(\mathbf{x}, \mathbf{x}) > 0$ вытекает, что $\lambda > 0$. Аналогично из $A \geqslant 0$ следует, что $\lambda \geqslant 0$.

Заметим, что для любой матрицы A и любой согласованной матричной нормы выполняется неравенство $\|A\| \geqslant |\lambda|$, где λ — собственное значение матрицы A. В самом деле, по определению собственного значения матрицы

$$A\mathbf{x} = \lambda\mathbf{x}$$

. По свойству согласованной матричной нормы $\|A\mathbf{x}\| \leqslant \|A\| \cdot \|\mathbf{x}\|$. Далее $\|\lambda\mathbf{x}\| = |\lambda|\|\mathbf{x}\|$. В итоге получаем $\|A\| \geqslant |\lambda|$.

1.9.2 Численные методы и линейная алгебра

Численные методы линейной алгебры — бурно развивающийся раздел численных методов. Приведём для подтверждения этого динамику числа научных публикаций за последние 200 лет:

1. с 1828 г. по 1974 г. (т.е. за 147 лет) — 4000 наименований;
2. с 1975 г. по 1980 г. (т.е. за 5 лет) — 3000;
3. с 1981 г. по 1984 г. (т.е. за 3 лет) — 4000.

Задачи линейной алгебры — это:

- решение систем линейных алгебраических уравнений (СЛАУ),
- вычисление определителей и обращение матриц,
- вычисление собственных значений и собственных векторов матриц.

1.9.3 Прямые методы решения СЛАУ

Пусть требуется найти решение системы

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = f_1, \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = f_2, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n = f_n, \end{cases} \tag{9.1}$$

или в компактной (векторной) форме

$$A\mathbf{x} = \mathbf{f}.$$

Мы будем считать, что $\Delta = \det A \neq 0$, то есть решение (9.1) существует и единственно.

В принципе, известны формулы Крамера, дающие в явной форме решение задачи (9.1):

$$x_i = \Delta_i/\Delta,$$

где Δ_i — определитель матрицы, которая получается из матрицы A заменой столбца с номером i столбцом правых частей (9.1). Определители при этом предлагается вычислять по формулам, рассматриваемым в курсах линейной алгебры. Например, для Δ:

$$\Delta = \sum_{\sigma} (-1)^{|\sigma|} a_{1\sigma(1)} a_{2\sigma(2)} \ldots a_{n\sigma(n)},$$

где σ — перестановка чисел $1, 2, \ldots n$, а $|\sigma|$ — чётность перестановки. Количество различных перестановок σ равно $n!$

Однако в качестве конкретного метода решения системы (9.1) данные формулы совершенно неприменимы, так как при подсчёте каждого определителя по приведённой выше формуле надо вычислить $n!$ слагаемых, что нереально при весьма умеренных n. Например, уже при $n = 100$ имеем $100! \gg 10^{90}$. Если одно слагаемое вычисляется, скажем за 10^{-7} сек, то время расчёта составит нереальное для практики значение

$$T \gg 10^{90} \cdot 10^{-7}\text{сек} = \frac{10^{83}}{86400}\,\text{суток} \approx 3 \cdot 10^{75}\,\text{лет}.$$

Метод Гаусса Метод состоит их *прямого* и *обратного* ходов.

В прямом ходе система уравнений с помощью элементарных преобразований строк матрицы приводится к верхнетреугольному виду. Не ограничивая существенно общности, рассмотрим работу прямого хода на примере системы трёх уравнений.

$$\left(\begin{array}{ccc|c} a_{11} & a_{12} & a_{13} & f_1 \\ a_{21} & a_{22} & a_{23} & f_2 \\ a_{31} & a_{32} & a_{33} & f_3 \end{array}\right) \to \left(\begin{array}{ccc|c} a_{11} & a_{12} & a_{13} & f_1 \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & f_2^{(1)} \\ 0 & a_{32}^{(1)} & a_{33}^{(1)} & f_3^{(1)} \end{array}\right) \to \left(\begin{array}{ccc|c} a_{11} & a_{12} & a_{13} & f_1 \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & f_2^{(1)} \\ 0 & 0 & a_{33}^{(2)} & f_3^{(2)} \end{array}\right)$$

Главный элемент. Обратимся к примеру системы (см. задачу 1.1 в главе «Практические занятия»)

$$\begin{cases} -10^{-7}x_1 + x_2 = 1, \\ \qquad x_1 + 2x_2 = 4. \end{cases}$$

Мы видели, что в одном из вариантов метода исключения результаты получались совершенно неверными. Напомним «механизм» возникновения больших погрешностей: деление на малые числа, появление больших (по величине) промежуточных результатов, потеря точности при вычитании больших (близких друг к другу) чисел.

Таким образом, порядок последовательного исключения неизвестных может сильно сказаться на результатах расчетов (тем более для систем высокого порядка такой исход весьма вероятен). Уменьшить опасность подобного рода, т. е. уменьшить в процессе выкладок вероятность деления на малые числа, позволяют варианты метода Гаусса с выбором *главного элемента.*

Выбор главного элемента по столбцам. Перед исключением x_1 отыскивается $\max\limits_i |a_{i1}|$. Допустим, максимум соответствует $i = i_0$. Тогда первое уравнение в исходной системе (9.1) меняем местами с i_0-м уравнением. (Для компьютера эта процедура связана с перестановкой двух строк расширенной матрицы (9.1).) После этого осуществляется первый шаг исключения. Затем перед исключением x_2 из оставшихся уравнений отыскивается $\max\limits_{2 \leqslant i \leqslant n} |a_{i2}^{(1)}|$ осуществляется соответствующая перестановка уравнений

и т.д.

Выбор главного элемента по строке. Перед исключением x_1 отыскивается $\max\limits_j |a_{1j}|\cdot$ Пусть максимум достигается при $j = j_0$. Тогда поменяем взаимно номера у неизвестных x_1 и x_{j_0} (максимальный по величине из коэффициентов 1-го уравнения окажется в позиции a_{11}) и приступим к процедуре исключения x_1, и т.д. Наиболее надежным является метод исключения с *выбором главного элемента по всей матрице* коэффициентов на каждом шаге исключения.

Рассмотренные модификации метода Гаусса позволяют, как правило, существенно уменьшить неблагоприятное влияние погрешностей округления на результаты расчета.

Впрочем, в прикладных задачах довольно часто приходится сталкиваться с линейными системами, при решении которых можно не заботиться о «вредном» воздействии неустранимых погрешностей на решение, спокойно применяя простейшую схему гауссова исключения (без выбора главного элемента). Это системы, для матриц которых выполнено *условие диагонального преобладания*:

$$|a_{ii}| > \sum_{j\neq i} |a_{ij}| \quad \text{для всех } i = \overline{1,n}.$$

Можно показать, что условие диагонального преобладания остается справедливым после каждого шага исключений в процессе приведения матрицы к треугольному виду, т.е.

$$|a_{ii}^{(k)}| > \sum_{\substack{j=k \\ j\neq i}}^{n} |a_{ij}^{(k)}| \quad i = \overline{k,n}$$

для всех $k = \overline{1,n-1}$. Это означает, что перед каждым исключением очередной неизвестной главный элемент будет находиться в «нужной позиции».

Количество арифметических операций зависит от вида исходной матрицы

I. Диагональная матрица.

$$\begin{pmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{pmatrix}, \qquad x_k = f_k/a_{kk}.$$

Для вычисления каждой переменной x_k, $k = 1, 2, \ldots, n$ требуется одно деление. Всего потребуется n операций деления.

II. Треугольная матрица.

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ 0 & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{pmatrix}, \quad \begin{array}{l} x_n = f_n/a_{nn}, \\ x_{n-1} = (f_{n-1} - a_{n-1,n}x_n)/a_{n-1,n-1}, \end{array}$$

Для нахождения x_n требуется одно деление. Для вычисления x_{n-1} требуется 3 операции (1 разность, 1 умножение, 1 деление). Для вычисления x_{n-2} требуется 5 операций и т.д. Всего потребуется $\sum\limits_{k=1}^{n}(2k-1) = n^2$ арифметических операций.

III. В общем случае, когда матрица имеет много ненулевых элементов, потребуется прямой и обратный ходы метода Гаусса. Точный расчёт говорит, что потребуется в общем случае $\frac{1}{6}n(n-1)(4n+7)$. Для больших $n \gg 10$ оценка приближённо равна $\frac{1}{6}n \cdot n \cdot 4n = \frac{2}{3}n^3$.

1.9.4 Погрешность численного решения СЛАУ

Погрешность входных данных

Оценим неустранимую погрешность решения СЛАУ. Источниками неустранимой погрешности являются не только округления при выполнении машинных операций, но также ошибки, содержащиеся в исходных данных. Разберёмся сначала с последними, предполагая, что арифметические операции выполняются точно.

Итак, пусть вместо системы

$$A\mathbf{x} = \mathbf{f} \tag{9.2}$$

решается задача

$$(A + \delta A)(\mathbf{x} + \delta\mathbf{x}) = \mathbf{f} + \delta\mathbf{f}. \tag{9.3}$$

Здесь δA — матрица возмущений, моделирующих ошибки коэффициентов исходной СЛАУ (9.2), $\delta\mathbf{f}$ — соответственно возмущения правых частей, $\delta\mathbf{x}$ — обусловленный этими возмущениями вектор «ошибок», отличающий решение (9.3) от решения (9.2).

Переписывая (9.3) в виде

$$A\mathbf{x} + A \cdot \delta\mathbf{x} + \delta A \cdot \mathbf{x} + \delta A \cdot \delta\mathbf{x} = \mathbf{f} + \delta\mathbf{f}$$

и вычитая из последнего соотношения (9.2), приходим к системе уравнений

$$A\delta\mathbf{x} + \delta A \cdot \delta\mathbf{x} = \delta\mathbf{f} - \delta A \cdot \mathbf{x}, \tag{9.4}$$

которая описывает зависимость $\delta\mathbf{x}$ от возмущений (ошибок) исходных данных.

Далее будем полагать, что возмущения коэффициентов уравнений δA и погрешности решения $\delta\mathbf{x}$ в достаточной мере малы так, что в уравнениях (9.4) можно пренебречь квадратичными членами $\delta A \cdot \delta\mathbf{x}$. Тогда интересующую нас ошибку $\delta\mathbf{x}$ можно представить в виде

$$\delta\mathbf{x} \approx A^{-1}(\delta\mathbf{f} - \delta A \cdot \mathbf{x}).$$

Вводя в рассмотрение нормы векторов и согласованные с ними нормы матриц, получим оценку величины погрешности:

$$\|\delta\mathbf{x}\| \approx \|A^{-1}(\delta\mathbf{f} - \delta A \cdot \mathbf{x})\| \leqslant \|A^{-1}\|(\|\delta\mathbf{f}\| + \|\delta A\| \cdot \|\mathbf{x}\|) = \\ = \|A^{-1}\| \cdot \left(\|\mathbf{f}\|\frac{\|\delta\mathbf{f}\|}{\|\mathbf{f}\|} + \|A\|\frac{\|\delta A\|}{\|A\|}\|\mathbf{x}\|\right).$$

Учитывая, что $\|\mathbf{f}\| = \|A\mathbf{x}\| \leqslant \|A\| \cdot \|\mathbf{x}\|$, получаем далее

$$\|\delta\mathbf{x}\| \leqslant \|A^{-1}\| \cdot \left(\|A\| \cdot \|\mathbf{x}\|\frac{\|\delta\mathbf{f}\|}{\|\mathbf{f}\|} + \|A\| \cdot \|\mathbf{x}\|\frac{\|\delta A\|}{\|A\|}\right). = \\ = \|A^{-1}\| \cdot \|A\| \cdot \|\mathbf{x}\| \left(\frac{\|\delta\mathbf{f}\|}{\|\mathbf{f}\|} + \frac{\|\delta A\|}{\|A\|}\right).$$

В итоге оценка для относительной погрешности решения может быть записана в виде

$$\frac{\|\delta\mathbf{x}\|}{\|\mathbf{x}\|} \lesssim \mu_A \left(\frac{\|\delta\mathbf{f}\|}{\|\mathbf{f}\|} + \frac{\|\delta A\|}{\|A\|} \right), \tag{9.5}$$

где $\mu_A = \|A^{-1}\| \cdot \|A\|$. Значение μ_A называется *числом обусловленности матрицы* A. Именно эта величина определяет, насколько сильно погрешности входных данных могут повлиять на решение системы (9.2).

Всегда $\mu_A \geqslant 1$. В самом деле, имеем $E = A^{-1}A$. Отсюда $1 = \|E\| = \|A^{-1}A\| \leqslant \|A^{-1}\| \cdot \|A\| = \mu_A$. Если значение μ_A является умеренным ($\mu_A \sim 1 \div 10$), ошибки входных данных слабо сказываются на решении; система (9.2) в этом случае называется *хорошо обусловленной*. Если μ_A велико ($\mu_A \geqslant 10^3$), система (9.2) *плохо обусловлена*, решение её сильно зависит от ошибок в правых частях и коэффициентах.

З а м е ч а н и е 1 . Вообще говоря, более точное представление о хорошей или плохой обусловленности системы должно опираться на требования, предъявляемые к решению. Если, к примеру, погрешность входных данных $\sim 10^{-6}$, а допустимая погрешность решения $\sim 10^{-2}$, то даже при $\mu \sim 10^4$ систему можно считать хорошо обусловленной.

З а м е ч а н и е 2 . Хотелось бы подчеркнуть, что данная свойство (обусловленность), выражаемое неравенством (9.5), никак не связано с предполагаемым методом решения системы, а является изначальной характеристикой решаемой задачи.

Пример. Рассмотрим систему

$$\begin{cases} 100x_1 + 99x_2 = 199, \\ 99x_1 + 98x_2 = 197. \end{cases}$$

Её решение $x_1 = x_2 = 1$.

Изменим теперь слегка её правые части

$$\begin{cases} 100x_1 + 99x_2 = 198{,}99, \\ 99x_1 + 98x_2 = 197{,}01. \end{cases}$$

Решение «искажённой» системы $x_1 \approx 2{,}97$, $x_2 \approx 0{,}99$.

Чтобы сопоставить полученные результаты с оценкой $\frac{||\delta\mathbf{x}||}{||\mathbf{x}||} \leqslant \mu_A \frac{||\delta\mathbf{f}||}{||\mathbf{f}||}$, будем пользоваться следующими согласованными нормами для векторов и матриц

$$||\mathbf{x}||_\infty = \max_i |x_i|, \quad ||A||_\infty = \max_i \sum_j |a_{ij}|.$$

Для рассмотреного примера имеем

$$\mathbf{f} = \begin{pmatrix} 199 \\ 197 \end{pmatrix}, \ \delta\mathbf{f} = \begin{pmatrix} -0,01 \\ 0{,}01 \end{pmatrix}, \ ||\mathbf{f}||_\infty = 199, \ ||\delta\mathbf{f}||_\infty = 0{,}01.$$

Относительная погрешность $\frac{||\delta\mathbf{f}||_\infty}{||\mathbf{f}||_\infty} \approx \frac{1}{2} \cdot 10^{-4} = 0{,}005\%$. Это очень малая величина.

Далее, $||A||_\infty = 199$,

$$\det A = -1, \ A^{-1} = \begin{pmatrix} -98 & 99 \\ 99 & -100 \end{pmatrix}, \ ||A^{-1}||_\infty = 199,$$

$$\mu_A = (199)^2 = 39601 \approx 4 \cdot 10^4.$$

Используя оценку, получим относительную погрешность решения $\frac{||\delta\mathbf{x}||_\infty}{||\mathbf{x}||_\infty} \leqslant 4 \cdot 10^4 \cdot \frac{10^{-4}}{2} = 2 = 200\%$. Это согласуется с результатами решения рассмотренных систем.

Погрешность округления при арифметических операциях

Можно показать, что полученное с помощью компьютера решение $\mathbf{x}' = \mathbf{x} + \delta\mathbf{x}$ СЛАУ, вычисленное методом Гаусса (с той или иной схемой выбора главного элемента или вообще без выбора), точно удовлетворяет уравнениям с определённым образом возмущёнными коэффициентами

$$(A + \delta A)(\mathbf{X} + \delta\mathbf{x}) = \mathbf{f}.$$

Для нормы матрицы так называемых эквивалентных возмущений δA справедлива оценка вида

$$\|\delta A\| \approx ng(A)\|A\|p^{-t}.$$

Здесь n — порядок системы, p — основание машинной арифметики (как правило, $p = 2$), t — число значащих цифр, учитываемых при выполнении

арифметических операций (для числа с плавающей запятой одинарной точности $t = 26$, для двойной: $t = 52$) $g(A) = \max\limits_k \dfrac{\max\limits_{i,j} |a_{ij}^{(k)}|}{\max\limits_{i,j} |a_{ij}|}$, где k — номер шага на этапе прямого хода метода исключений. (Таким образом, величина $g(A)$ показывает, насколько могут возрасти пересчитываемые элементы матрицы на стадии приведения её к треугольному виду. Отсюда её название — *коэффициент роста.*)

Привлекая далее (9.5), получаем оценку ошибок решения системы (9.2) за счёт погрешностей вычислений:

$$\frac{\|\delta \mathbf{x}\|}{\|\mathbf{x}\|} \approx \mu_A n g(A) p^{-t}. \tag{9.6}$$

Если, например, решается система из 10^3 уравнений и $\mu_A \approx 1$, $g(A) \approx 1$, то при счёте с двойной точностью ($p = 2$, $t = 52$) нельзя рассчитывать на точность лучшую, нежели $\dfrac{\|\delta \mathbf{x}\|}{\|\mathbf{x}\|} \sim n \cdot 2^{-52} \sim n \cdot 10^{-15}|_{n=10^3} \sim 10^{-12}$. Если при этом $\mu_A \approx 10^{12}$, то может произойти полная потеря точности.

З а м е ч а н и е . Плохо обусловленные СЛАУ вызывают определённые трудности при решении. Из (9.5) следует, что решение их сильно зависит от ошибок входных данных, а из (9.6) вытекает, что даже при отсутствии ошибок во входных величинах может произойти значительная (если не полная) потеря точности на стадии вычислений по методу Гаусса за счёт погрешностей округлений.

1.9.5 Итерационные методы решения СЛАУ

По-прежнему будем рассматривать системы вида $A\mathbf{x} = \mathbf{f}$. Различные варианты итерационных методов связаны с переходом к эквивалентной системе

$$A\mathbf{x} = \mathbf{f} \quad \Leftrightarrow \quad \mathbf{x} = P\mathbf{x} + \mathbf{g}.$$

Берём некоторое начальное приближение $\mathbf{x}^{(0)}$ и очевидным образом строим итерационный процесс $\mathbf{x}^{(k)} = P\mathbf{x}^{(k-1)} + \mathbf{g}$. Здесь $\mathbf{x}^{(k)}$ обозначает k-е по счёту приближение к искомому вектору $\mathbf{x}$.

Условия сходимости метода последовательных приближений формулируются в следующих теоремах.

Теорема 9.2. *Для сходимости итераций*

$$\mathbf{x}^{(k)} = P\mathbf{x}^{(k-1)} + \mathbf{g}, \text{ где } \mathbf{x}^{(0)} \text{ задано} \tag{9.7}$$

к решению системы

$$\mathbf{x} = P\mathbf{x} + \mathbf{g} \tag{9.8}$$

достаточно, чтобы в какой-либо норме выполнялось условие

$$||P|| \leqslant q < 1.$$

Тогда независимо от выбора $\mathbf{x}^{(0)}$

$$||\mathbf{x}^{(k)} - \mathbf{x}^*|| \leqslant q^k ||\mathbf{x}^{(0)} - \mathbf{x}^*||,$$

где $\mathbf{x}^*$ *— точное решение.*

Доказательство. Подстановка точного решения в (9.8) обращает последнее в тождество

$$\mathbf{x}^* = P\mathbf{x}^* + \mathbf{g}.$$

Вычитая его из (9.7), получим

$$\mathbf{x}^{(k)} - \mathbf{x}^* = P(\mathbf{x}^{(k-1)} - \mathbf{x}^*),$$

где $(\mathbf{x}^{(k)} - \mathbf{x})$ — вектор погрешности (или просто погрешность k-го решения).

Оценивая погрешность по какой-либо норме (с которой согласована норма матрицы, фигурирующая в условии теоремы), получаем

$$||\mathbf{x}^{(k)} - \mathbf{x}^*|| \leqslant ||P|| \cdot ||\mathbf{x}^{(k-1)} - \mathbf{x}^*|| \leqslant$$
$$\leqslant q||\mathbf{x}^{(k-1)} - \mathbf{x}^*|| \leqslant \ldots \leqslant q^k ||\mathbf{x}^{(0)} - \mathbf{x}^*||.$$

Очевидно, что при $q < 1$ $\lim\limits_{k\to\infty} \mathbf{x}^{(k)} = \mathbf{x}^*$. □

Теорема 9.3. *(без доказательства) Для сходимости итераций* (9.7) *к решению системы* (9.8) *необходимо и достаточно, чтобы все собственные значения матрицы* P *по абсолютной величине были меньше единицы.*

Выбор метода Теперь самое время осознать, зачем, собственно, нужны итерационные методы, если мы умеем вычислять решение, пользуясь, например, какой-либо модификацией метода Гаусса. Вопрос становится ясным, если оценить эффективность различных подходов с точки зрения вычислительных затрат.

Метод Гаусса (в простейшей интерпретации), как мы видели, при $n \gg 1$ требует выполнения приблизительно $\frac{2}{3}n^3$ арифметических операций. Метод итераций $\mathbf{x}^{(k)} = P\mathbf{x}^{(k-1)} + \mathbf{g}$ реализуется приблизительно за $(2n^2)K$ операций ($2n^2$ умножений и сложений связано с умножением матрицы P на вектор $\mathbf{x}^{(k-1)}$, K — число приближений). Если допустимая погрешность достигается при $K < n/3$, то метод итераций становится предпочтительней. В задачах, с которыми практически приходится иметь дело, зачастую $K \ll n$.

Кроме того, итерационные методы могут оказаться предпочтительней с точки зрения устойчивости вычислений, в смысле влияния вычислительных погрешностей на результаты расчетов.

Метод Якоби. Запишем каждое уравнение системы $A\mathbf{x} = \mathbf{f}$ в виде, разрешённом относительно неизвестного с коэффициентом на главной диагонали матрицы A:

$$x_m = \frac{1}{a_{mm}}(f_m - a_{m1}x_1 - a_{m,m-1}x_{m-1} - a_{m,m+1}x_{m+1} - \ldots - a_{mn}x_n),$$

$m = 1, 2, \ldots, n$.

То есть мы переписали $A\mathbf{x} = \mathbf{f}$ в виде $\mathbf{x} = P\mathbf{x} + \mathbf{g}$ с матрицей

$$P = -\begin{pmatrix} 0 & \frac{a_{12}}{a_{11}} & \frac{a_{13}}{a_{11}} & \ldots & \frac{a_{1n}}{a_{11}} \\ \frac{a_{21}}{a_{22}} & 0 & \frac{a_{23}}{a_{22}} & \ldots & \frac{a_{2n}}{a_{22}} \\ \vdots & & & & \vdots \\ \frac{a_{n1}}{a_{nn}} & \frac{a_{n2}}{a_{nn}} & \frac{a_{n3}}{a_{nn}} & \ldots & 0 \end{pmatrix}.$$

Если ввести в рассмотрение диагональную матрицу

$$D = \begin{pmatrix} a_{11} & & & 0 \\ & a_{22} & & \\ & & \ddots & \\ 0 & & & a_{nn} \end{pmatrix},$$

то $P = -D^{-1}(A - D)$, $\mathbf{g} = D^{-1}\mathbf{f}$.

Итерационный процесс (9.7) с определённой таким образом матрицей P называется *методом Якоби*. Фактически вычисления проводятся по формулам

$$x_m^{(k)} = \frac{1}{a_{mm}}(f_m - a_{m1}x_1^{(k-1)} - \ldots - a_{m,m-1}x_{m-1}^{(k-1)} - \\ - a_{m,m+1}x_{m+1}^{(k-1)} - \ldots - a_{mn}x_n^{(k-1)}),\ m = 1, 2, \ldots, n. \quad (9.9)$$

Для сходимости метода Якоби достаточно, чтобы для исходной матрицы A имело место диагональное преобладание, т.е. чтобы коэффициенты исходных уравнений удовлетворяли условиям

$$|a_{ii}| > \sum_{j \neq i} |a_{ij}| \quad \text{для всех } i.$$

В самом деле, тогда условие теоремы 9.2 выполнено для нормы $||P||_\infty$:

$$||P||_\infty = \max_i \sum_{j \neq i} \frac{|a_{ij}|}{|a_{ii}|} = \max_i \frac{\sum\limits_{j \neq i} |a_{ji}|}{|a_{ii}|} < 1.$$

Метод Зейделя.

Этот метод отличается от метода Якоби только тем, что при вычислении k-го приближения m-й компоненты используются уже вычисленные k-е

приближения предыдущих (1-й, 2-й, ..., $(m-1)$-й) компонент:

$$
\begin{aligned}
x_1^{(k)} &= \frac{1}{a_{11}}\left(f_1 - a_{12}x_2^{(k-1)} - a_{13}x_3^{(k-1)} - \ldots - a_{1n}x_n^{(k-1)}\right),\\
x_2^{(k)} &= \frac{1}{a_{22}}\left(f_2 - a_{21}x_1^{(k)} - a_{23}x_3^{(k-1)} - \ldots - a_{2n}x_n^{(k-1)}\right),\\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\
x_m^{(k)} &= \frac{1}{a_{mm}}\Big(f_m - a_{m1}x_1^{(k)} - a_{m2}x_2^{(k)} - \ldots - a_{mm-1}x_{m-1}^{(k)} -\\
&\qquad\qquad -a_{mm+1}x_{m+1}^{(k-1)} - \ldots - a_{mn}x_n^{(k-1)}\Big),\\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots
\end{aligned}
$$

Если представить матрицу A в виде суммы $A = A_- + D + A_+$ (A_- — нижняя треугольная, A_+ — верхняя треугольная и D — диагональная матрицы с элементами исходной матрицы A), то методу Зейделя соответствует матрица

$$P = -(A - A_+)^{-1}A_- = -(A_- + D)^{-1}A_-.$$

Можно доказать, что метод Зейделя гарантировано сходится, если:

- выполнено условие диагонального преобладания матрицы A; или
- матрица A является симметричной и положительно определенной.

В одинаковых условиях (при наличии диагонального преобладания) метод Зейделя сходится примерно в два раза быстрее метода Якоби.

Однопараметрический метод итераций

Перепишем $A\mathbf{x} = \mathbf{f}$ в виде $\mathbf{x} = \mathbf{x} - \tau(A\mathbf{x} - \mathbf{f}) = (E - \tau A)\mathbf{x} + \tau\mathbf{f}$, где τ — пока неопределенный параметр.

Мы фактически привели $A\mathbf{x} = \mathbf{f}$ к форме $\mathbf{x} = P\mathbf{x} + \mathbf{g}$, где $P = E - \tau A$ и $\mathbf{g} = \tau\mathbf{f}$. Итерационная последовательность имеет вид

$$\mathbf{x}^{(k)} = P\mathbf{x}^{(k-1)} + \mathbf{g}, \tag{9.10}$$

Далее будем предполагать, что матрица исходной системы симметрична и положительно определена (т.е. $A^\top = A$ и $A > 0$) и что известны

границы спектра матрицы A (минимальное и максимальное собственные значения). При этих предположениях мы не только определим диапазон значений τ, гарантирующих сходимость, но и найдем оптимальное τ, при котором величина погрешности приближений убывает с номером приближения наиболее быстро.

Итак, замечая, что

$$\mathbf{x}^* = P\mathbf{x}^* + \mathbf{g}, \tag{9.11}$$

$\mathbf{x}^*$ — точное решение $A\mathbf{x} = \mathbf{f}$, и вводя в рассмотрение вектор ошибки k-го приближения $\mathbf{r}^{(k)} = \mathbf{x}^{(k)} - \mathbf{x}^*$, получим, вычитая (9.11) из (9.10):

$$\mathbf{r}^{(k)} = P\mathbf{r}^{(k-1)}. \tag{9.12}$$

Собственные значения матрицы A обозначим через λ_i, а собственные значения P — через μ_i. Очевидно, что $\mu_i = 1 - \tau\lambda$.

Найдём Евклидову норму P. Так как $E = E^\top$ и $A = A^\top$, то $P = E - \tau A = E^\top - \tau A^\top = P^\top$. Если μ_i — собственное значение матрицы P, то μ_i^2 — будет собственным значением матрицы $P^\top P = P^2$. Действительно из $P\mathbf{x} = \mu\mathbf{x}$ следует, что $P^2\mathbf{x} = \mu P\mathbf{x} = \mu^2\mathbf{x}$. Для подчинённой нормы 2 справедливо

$$\|P\|_2 = \sqrt{\max_i \mu_i^2} = \max_i |\mu_i|.$$

По теореме 9.2 для сходимости итерационного процесса требуется, чтобы выполнялось $\|P\|_2 = \max\limits_i |\mu_i| \leqslant q < 1$. Из этой же теоремы следует, что

$$\|\mathbf{r}^{(k)}\|_2^2 \leqslant \max_i \mu_i^2 \cdot \|\mathbf{r}^{(k-1)}\|_2^2. \tag{9.13}$$

Таким образом, если $\max\limits_i |\mu_i| \leqslant q < 1$, то погрешность будет убывать с номером приближения как член геометрической прогрессии со знаменателем q.

Оптимизация параметра. В силу (9.13) для быстрого убывания нормы ошибки $\|\mathbf{r}^{(k)}\|$ нужно, чтобы величина $\max\limits_i |\mu_i|$ была как можно меньше. Кроме того для сходимости итерационного процесса необходимо, чтобы $\max\limits_i |\mu_i| \leqslant q < 1$.

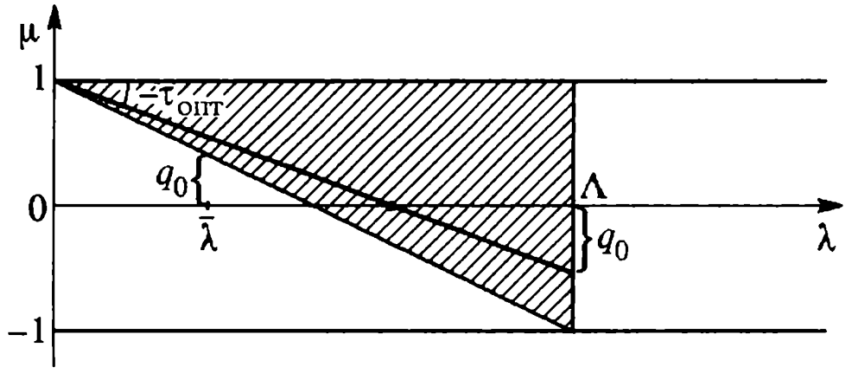


Рис. 9.1: Выбор оптимального значения параметра τ.

Для удобства введём следующие обозначения

$$\bar{\lambda} = \min_i \lambda_i, \qquad \Lambda = \max_i \lambda_i.$$

На рис. 9.1 в системе координат λ и μ прямая

$$\mu = 1 - \tau\lambda \tag{9.14}$$

проходит через точку $(\lambda, \mu) = (0, 1)$. Так как $\max\limits_i |\mu_i| < 1$, то параметр τ может меняться в пределах $0 < \tau < 2/\Lambda$. Следовательно, прямая (9.14) при различных τ описывает заштрихованную на рис. 9.1 область. Наибольшие отклонения прямой $\mu = 1 - \tau\lambda$ от нуля на отрезке $\bar{\lambda} \leqslant \lambda \leqslant \Lambda$ будут при $\lambda = \bar{\lambda}$ или при $\lambda = \Lambda$. Из геометрических соображений очевидно, что значение $\max\limits_i |\mu_i| = \max\limits_i |1 - \tau\lambda_i|$ будет достигнуто либо при $\lambda = \bar{\lambda}$, либо при $\lambda = \Lambda$.

Вспомним, что наша цель — это подобрать такое значение τ, чтобы число $\max\limits_i |\mu_i|$ было как можно меньше. Глядя на рисунок, несложно заметить, что при искомом оптимальном значении τ прямая $\mu = 1 - \tau\lambda$ должна пересечь с е р е д и н у отрезка $[\bar{\lambda}, \Lambda]$, то есть

$$\mu = 0 = 1 - \tau \cdot \underbrace{\frac{\bar{\lambda} + \Lambda}{2}}_{\text{сер. отр. } [\bar{\lambda}, \Lambda]}, \quad \text{откуда} \quad \tau_{\text{опт}} = \frac{2}{\bar{\lambda} + \Lambda}.$$

Имеем

$$q = \max_i |\mu_i| = |1 - \tau_{\text{опт}}\bar{\lambda}| = |1 - \tau_{\text{опт}}\Lambda| = 1 - \frac{2\bar{\lambda}}{\bar{\lambda} + \Lambda} = \frac{\bar{\lambda} - \Lambda}{\bar{\lambda} + \Lambda}. \tag{9.15}$$

Именно величина q определяет реальный темп убывания погрешности с номером приближения в рамках оптимального однопараметрического итерационного процесса. Имеет место оценка

$$\|\mathbf{r}^{(k)}\| \leqslant q^k \|\mathbf{r}^{(0)}\|. \tag{9.16}$$

Откуда, получаем априорную оценку числа приближений, гарантирующих достижение заданной точности ε:

$$k \geqslant k_0 = \frac{\ln \frac{\varepsilon}{\|\mathbf{r}^{(0)}\|}}{\ln q_0}$$

Найдем μ_A, используя норму 2. По условию матрица A — симметричная, то есть $A^\top = A$. Тогда $A^\top A = A^2$. Если λ_i — собственное значение матрицы A, то λ_i^2 — собственное значение матрицы $A^2 = A^\top A$. Для нормы A получим

$$\|A\|^2 = \sqrt{\max_i |\lambda_i^2|} = \max_i |\lambda_i| \overset{A>0}{=} \max_i \lambda_i = \Lambda.$$

Аналогично для обратной матрицы с помощью леммы 9.1

$$\|A^{-1}\|_2 = \sqrt{\max_i \frac{1}{|\lambda_i^2|}} = \max_i \frac{1}{|\lambda_i|} \overset{A>0}{=} \max_i \frac{1}{\lambda_i} = \frac{1}{\bar{\lambda}}.$$

В результате

$$\mu_A = \Lambda / \bar{\lambda}.$$

Теперь можно в (9.15) выразить q через μ_A

$$q = \frac{\mu_A - 1}{\mu_A + 1}.$$

Если $\mu_A \gg 1$ (число обусловленности велико), то

$$\ln q_0 = \ln \frac{\mu_A - 1}{\mu_A + 1} = \ln(\mu_A - 1) - \ln(\mu_A + 1) \approx -2/\mu_A$$

и

$$k_0 \approx \frac{\mu_A}{2} \ln \frac{\varepsilon}{\|\mathbf{r}^{(0)}\|}. \tag{9.17}$$

З а м е ч а н и е . Здесь приводятся некоторые оценки для систем с большим числом обусловленности. Это не случайно. Дело в том, что с такими

системами приходится иметь дело довольно часто при численном решении уравнений с частными производными.

П р и м е р . Пусть надо решить систему из $n = 10^4$ линейных уравнений с симметричной положительно определенной матрицей. Метод Гаусса требует в этом случае выполнения порядка $n^3 \sim 10^{12}$ элементарных операций. В итерационных методах для перехода от одного приближения к следующему необходимо выполнить порядка n^2 операций. Таким образом, объем вычислений, который сопряжен с методом итераций с одним оптимальным параметром ($k_0 n^2$), согласно (9.17) порядка $\mu_A \cdot \ln \frac{1}{\varepsilon} \cdot 10^8$, и при не слишком больших числах обусловленности этот метод эффективнее метода Гаусса.

Каноническая форма записи итерационных методов

Многообразие итерационных схем, созданных для решения различных линейных систем, можно представить в канонической форме записи:

$$B_{k+1}\frac{\mathbf{X}^{(k+1)} - \mathbf{X}^{(k)}}{\tau_{k+1}} + A\mathbf{X}^{(k)} = \mathbf{f}.$$

Принята следующая классификация:

- $B_k = E$ — *явные* итерационные процессы,
- $B_k \neq E$ — *неявные* итерационные процессы,
- $B_k = B$, $\tau_k = \tau$ — *стационарные* итерационные процессы.

Нами были рассмотрены следующие частные случаи:

1. $B_k = E$, $\tau_k = \tau$ — однопараметрический метод;
2. $B_k = D$, $\tau_k = 1$ — метод Якоби;
3. $B_k = D + A_-$, $\tau_k = 1$ — метод Зейделя.

1.10 Метод Прогонки.

В общем случае системы с трёхдиагональной матрицей имеют вид

$$\begin{cases} a_j y_{j-1} - c_j y_j + b_j y_{j+1} = -f_j, \quad j = 1, 2, \ldots, n-1, \\ y_0 = \varkappa_1 y_1 + \mu_1, \quad y_n = \varkappa_2 y_{n-1} + \mu_2. \end{cases} \tag{10.1}$$

Или в матричном виде $A\mathbf{y} = \mathbf{f}$:

$$\underbrace{\begin{pmatrix} -1 & \varkappa_1 & & & & 0 \\ a_1 & -c_1 & b_1 & & & \\ & a_2 & -c_2 & b_2 & & \\ & & \ddots & \ddots & \ddots & \\ & & & a_{n-1} & -c_{n-1} & b_{n-1} \\ 0 & & & & \varkappa_2 & -1 \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_{n-1} \\ y_n \end{pmatrix}}_{\mathbf{y}} = \underbrace{\begin{pmatrix} -\mu_1 \\ -f_1 \\ -f_2 \\ \vdots \\ -f_{n-1} \\ -\mu_2 \end{pmatrix}}_{\mathbf{f}}$$

В матрице A на главной диагонали и в векторе $\mathbf{f}$ стоят элементы со знаком «минус». Это объясняется применением метода прогонки для решения разностных схем для дифференциальных уравнений второго порядка.[6]

Будем искать решение в виде

$$y_j = \alpha_{j+1} y_{j+1} + \beta_{j+1}, \quad j = 0, 1, \ldots, n-1, \tag{10.2}$$

где α_{j+1}, β_{j+1} — неизвестные пока коэффициенты. Отсюда найдём

$$y_{j-1} = \alpha_j y_j + \beta_j = \alpha_j \underbrace{(\alpha_{j+1} y_{j+1} + \beta_{j+1})}_{y_j} + \beta_j = \alpha_j \alpha_{j+1} y_{j+1} + (\alpha_j \beta_{j+1} + \beta_j),$$

где $j = 1, 2, \ldots, n-1$. Подставляя полученные коэффициенты для y_j и y_{j-1} в уравнение (10.1) и объединяя слагаемые с y_{j+1}, приходим при $j = 1, 2, \ldots, n-1$ к уравнению

$$\underbrace{[\alpha_{j+1}(a_j \alpha_j - c_j) + b_j]}_{\text{приравняем к нулю}} y_{j+1} + \underbrace{[\beta_{j+1}(a_j \alpha_j - c_j) + a_j \beta_j + f_j]}_{\text{приравняем к нулю}} = 0.$$

Последнее уравнение будет выполнено, если коэффициенты α_{j+1}, β_{j+1} выбрать такими, чтобы выражения в квадратных скобках обращались в

[6] Например, вторая производная в точке x_k аппроксимируется разделённой разностью $\frac{1}{h^2}(y_{k-1} - 2y_k + y_{k+1})$, где коэффициент при y_k отрицательный.

нуль. А именно, достаточно положить

$$\alpha_{j+1} = \frac{b_j}{c_j - \alpha_j a_j}, \quad \beta_{j+1} = \frac{a_j \beta_j + f_j}{c_j - \alpha_j a_j}, \quad j = 1, 2, \ldots, n-1. \tag{10.3}$$

Для того, чтобы применять последние соотношения нужно задать начальные значения α_1, β_1. С одной стороны $y_0 = \alpha_1 y_1 + \beta_1$, с другой стороны по условию задано, что $y_0 = \varkappa_1 y_1 + \mu_1$. Таким образом, получаем

$$\alpha_1 = \varkappa_1, \quad \beta_1 = \mu_1. \tag{10.4}$$

Нахождение коэффициентов α_{j+1}, β_{j+1} по формулам (10.3) и (10.4) называется *прямой прогонкой*. После того как прогоночные коэффициенты α_{j+1}, β_{j+1}, $j = 0, 1, \ldots n-1$, найдены, решение системы (10.1) находится по рекуррентной формуле (10.2), начиная с $j = n-1$. Для начала счёта по этой формуле требуется знать y_n, которое определяется из уравнений

$$y_n = \varkappa_2 y_{n-1} + \mu_2, \quad y_{n-1} = \alpha_n y_n + \beta_n$$

и равно $(\varkappa_2 \beta_n + \mu_2)/(1 - \varkappa_2 \alpha_n)$. Нахождение y_j по формулам

$$y_j = \alpha_{j+1} y_{j+1} + \beta_{j+1}, \quad j = n-1, n-2, \ldots, 0,$$

$$y_n = \frac{\varkappa_2 \beta_n + \mu_2}{1 - \varkappa_2 \alpha_n} \tag{10.5}$$

называется *обратной прогонкой*.

Метод прогонки можно применять, если знаменатели выражений (10.3), (10.5) не обращаются в нуль. Достаточные для этого условия перечислены в следующих двух теоремах.

Теорема 10.1 (достаточное условие применимости прогонки)**.** *Пусть*

$$a_j \neq 0, \quad b_j \neq 0,$$

$$|c_j| \geqslant |a_j| + |b_j|, \ j = 1, 2, \ldots, n-1 \ \textit{(диагональное преобладание)}, \tag{10.6}$$

$$|\varkappa_1| \leqslant 1, \ |\varkappa_2| < 1, \tag{10.7}$$

тогда метод прогонки применим.

Доказательство. Сначала докажем по индукции, что при условиях теоремы $|\alpha_j| \leqslant 1$, $j = 1, \ldots, n-1$. Базис индукции: $\alpha_1 = \varkappa_1$ и $|\varkappa_1| \leqslant 1$, следовательно, $|\alpha_1| \leqslant 1$. Индуктивный переход: пусть $|\alpha_j| \leqslant 1$, докажем, что $|\alpha_{j+1}| \leqslant 1$. Из оценок[7]

$$|c_j - \alpha_j a_j| \geqslant \big||c_j| - |\alpha_j|\,|a_j|\big| \overset{|\alpha_j| \leqslant 1}{\geqslant} \big||c_j| - |a_j|\big| \overset{(10.6)}{\geqslant} |b_j| > 0,$$

т.е. знаменатели выражений (10.3) не обращается в нуль. Более того,

$$|\alpha_{j+1}| = \frac{|b_j|}{|c_j - \alpha_j a_j|} \leqslant 1.$$

Следовательно, $|\alpha_j| \leqslant 1$, $j = 1, 2, \ldots, n$. Далее, учитывая условие теоремы $|\varkappa_2| < 1$ и только что доказанное $|\alpha_n| \leqslant 1$, имеем

$$|1 - \varkappa_2 \alpha_n| \geqslant 1 - |\varkappa_2|\,|\alpha_n| \geqslant 1 - |\varkappa_2| > 0,$$

т.е. не обращается в нуль и знаменатель в выражении для y_n. □

Теорема 10.2 (альтернативный вариант)**.** *Пусть* $a_j \neq 0$, $b_j \neq 0$, $|c_j| > |a_j| + |b_j|$, $j = 1, 2, \ldots, n-1$, $|\varkappa_1| \leqslant 1$, $|\varkappa_2| \leqslant 1$, *тогда метод прогонки применим.*

Доказательство. Действуем аналогично предыдущему доказательству. В данном случае из предположения $|\alpha_j| \leqslant 1$ следует

$$|c_j - \alpha_j a_j| \geqslant \big||c_j| - |a_j|\big| > |b_j|, \quad |\alpha_{j+1}| < 1,$$

т.е. все прогоночные коэффициенты, начиная со второго, по модулю строго меньше единицы. При этом $|1 - \varkappa_2 \alpha_n| \geqslant 1 - |\varkappa_2|\,|\alpha_n| \geqslant 1 - |\alpha_n| > 0$. □

Количество арифметических операций у метода прогонки оценивается $\sim 8n$. Это очень мало сравнительно с другими методами решения СЛАУ. Причина кроется в том, что матрица A содержит много нулевых элементов.

[7] Использованы неравенства «треугольника»: для любых $a, b \in \mathbb{R}$ справедливо, что $|a| + |b| \geqslant |a + b|$ и $|a - b| \geqslant \big||a| - |b|\big|$.

Устойчивость метода прогонки к погрешностям входных данных. Если выполняются условия теоремы 10.1 или теоремы 10.2, то, как было доказано, $|\alpha_j| \leqslant 1$, $j = 1, 2, \ldots, n$. Пусть на каком-либо шаге вычислений была внесена погрешность. Тогда эта погрешность не будет возрастать при переходе к следующим шагам. Действительно, пусть в формулах (10.2) или (10.5) при $j = j_0 + 1$ вместо y_{j_0+1} вычислена величина $\widetilde{y}_{j_0+1} = y_{j_0+1} + \delta_{j_0+1}$. Тогда на следующем шаге вычислений, т.е. при $j = j_0$, вместо $y_{j_0} = \alpha_{j_0+1} y_{j_0+1} + \beta_{j_0+1}$ получим величину $\widetilde{y}_{j_0} = \alpha_{j_0+1}(y_{j_0+1} + \delta_{j_0+1}) + \beta_{j_0+1}$ и погрешность окажется равной

$$\delta_{j_0} = \widetilde{y}_{j_0} - y_{j_0} = \alpha_{j_0+1}\delta_{j_0+1}.$$

Отсюда получим, что $|\delta_{j_0}| \leqslant |\alpha_{j_0+1}|\,|\delta_{j_0+1}| \leqslant |\delta_{j_0+1}|$, т.е. погрешность не возрастает.

1.11 Численное решение дифференциальных уравнений

Будем рассматривать обыкновенные дифференциальные уравнения

$$F(x, u(x), u'(x), \ldots, u^{(n)}) = 0. \tag{11.1}$$

Как известно, в общем случае такие уравнения имеют бесконечно много решений. В приложениях требуется выделить одно из этих решений. Поэтому, часто уравнение (11.1) рассматривают в сочетании с дополнительными ограничивающими условиями. Рассмотрим основные типы дополнительных условий, с которыми мы будем дальше работать.

1. *Задача Коши* для дифференциального уравнения первого порядка
$$\begin{cases} u' = f(x, u),\ x \in [a, b], \\ u(x_a) = u_a. \end{cases} \tag{11.2}$$

2. *Краевая задача* для линейного дифференциального уравнения второго порядка
$$\begin{cases} u'' + g(x)u' + h(x)u = f(x),\ x \in [a, b], \\ \alpha_1 u'(a) + \beta_1 u(a) = u_a, \\ \alpha_2 u'(b) + \beta_2 u(b) = u_b. \end{cases} \tag{11.3}$$
Граничные условия принято разделять на следующие типы: (а) *первого рода*, если $\alpha_i = 0$, $i = 1, 2$; (б) *второго рода*, если $\beta_i = 0$, $i = 1, 2$; (в) *третьего рода*, если α_i и β_i одновременно отличны от нуля.

При исследовании численных методов для каждой из перечисленных задач будем заранее предполагать, что решение соответствующей задачи существует, единственно и обладает необходимыми свойствами гладкости.

Многие методы решения дифференциальных задач сводятся к следующему. На интересующем нас отрезке $[a, b]$, на котором требуется найти численное решение задачи для дифференциального уравнения, вводится набор точек или *сетка*

$$\omega = \{x_0, x_1, \ldots, x_n\}.$$

В дальнейшем мы всегда будем считать, что расстояние между соседними узлами сетки x_k и x_{k+1} есть константа h, называемая шагом сетки.

Кроме того полагаем $x_0 = a$, $x_n = b$. Теперь исходное дифференциальное уравнение мы будем рассматривать только в узлах сетки ω

$$F(x_k, u(x_k), u'(x_k), u''(x_k), \ldots, u^{(p)}(x_k)) = 0,\ k = 0, 1, \ldots, n. \qquad (11.4)$$

Для краткости договоримся вместо $u(x_k)$ писать u_k. Теперь осталось перейти от дифференциального уравнения (11.4) к разностному. Для этого заменим все вхождения символа u_k на y_k, а все вхождения символов производной на соответствующие разностные производные. Например,

$$u'_k \to \frac{1}{h}(y_{k+1} - y_k) \text{ или } u'_k \to \frac{1}{2h}(y_{k+1} - y_{k-1}),$$

$$u''_k \to \frac{1}{h^2}(y_{k-1} - 2y_k + y_{k+1}).$$

Переход от символов производной к разностным производным неоднозначен. При выборе той или иной разностной формулы могут сыграть роль порядок погрешности формулы (например, $O(h)$ или $O(h^2)$), а также число соседних узлов, задействованных в формуле (например, два узла x_k и x_{k+1} или три узла x_{k-1} и x_{k+1}).

Решением разностного уравнения будет *сеточная функция* $y_k = y(x_k)$. Термин «сеточная» говорит от том, что область определения функции $y(x)$ есть не весь отрезок $[a, b]$, а только узлы сетки ω. Итак, $u(x)$ и $y(x)$ есть различные функции, но при определённых условиях можно считать $u_k \approx y_k$, $k = 0, 1, \ldots, n$.

При использовании приближённых методов основным является вопрос о сходимости. Понятие сходимости приближённого метода можно сформулировать по-разному. Будем рассматривать понятие *сходимости при* $h \to 0$. Оно означает следующее. Фиксируем точку x и построим последовательность сеток ω_h таких, что $h \to 0$ и $x_n = nh = x$ (при этом, очевидно, $n \to \infty$). Говорят, что численное решение *сходится в точке* x к точному решению, если $|y_n - u(x_n)| \to 0$ при $h \to 0$, $x_n = x$.

Численное решение *сходится на отрезке* $[a, b]$ к точному решению, если оно сходится в каждой точке этого отрезка.

Погрешность решения — это числовая характеристика δ_h, показывающая, насколько истинное решение $u(x_k)$ дифференциального уравнения

отклоняется от решения y_k системы разностных уравнений. Для погрешности решения мы будем использовать величину $\delta_h = \max\limits_k |y_k - u(x_k)| = ||y_k - u(x_k)||_\infty$, $k = 0, 1, \ldots, n$.

Говорят, что численное решение имеет p-ый порядок точности, если существует число $p > 0$ такое, что $\delta_h = O(h^p)$ при $h \to 0$.

Очевидно, в общем случае точное решение дифференциальной задачи $u(x)$ и решение разностной схемы y_k, $k = 0, 1, \ldots, n$ не совпадают. Поэтому, если решение дифференциального уравнения подставить в разностное уравнение, последнее перестанет быть равенством из-за появления невязки. Данная невязка называется *погрешностью аппроксимации* и обозначается ψ. Если при уменьшении шага $h \to 0$ выполняется $\psi = O(h^p)$ для некоторого p, то говорят, что погрешность аппроксимации имеет порядок p. Если $\psi_k \not\to 0$ при $h \to 0$, то говорят, что разностная схема *не аппроксимирует* дифференциальное уравнение.

Найти аналитически погрешность решения — это часто очень сложная или неразрешимая задача. Напротив, вычислить погрешность аппроксимации довольно легко. На практике важно знать, какой порядок точности имеет та или иная разностная схема. Оказывается, что ответить на этот вопрос можно, зная порядок погрешности аппроксимации.

Теорема 11.1 (без доказательства)**.** *Пусть а) некоторая разностная схема аппроксимирует задачу (11.2) или задачу (11.3), причём $\psi_k = O(h^p)$; б) решение разностной схемы y_k, $k = 0, 1, \ldots, n$ устойчиво к погрешностям входных данных (т.е. малые погрешности в функции f из правой части (11.2) или (11.3) приводят к незначительным изменениям решения y_k). Тогда решение y_k, $k = 0, 1, \ldots, n$ разностной схемы сходится к решению $u(x_k)$ дифференциального уравнения при $h \to 0$, и имеем место следующая оценка погрешности $\delta_h = O(h^p)$.*

Мы не будем останавливаться на вопросе устойчивости решения y_k, $k = 0, 1, \ldots, n$ разностных схем. Отметим лишь, что все рассматриваемые далее методы устойчивы.

Метод Эйлера. Заменим в задаче Коши (11.2) производную правой разделённой разностью

$$\frac{y_{k+1}-y_k}{h} = f(x_k, y_k),\ k = 0, 1, 2, \ldots, n-1,\ y_0 = u_a. \tag{11.5}$$

Решение этой системы уравнений находится явным образом по рекуррентной формуле

$$y_{k+1} = y_k + hf(x_k, y_k),\ k = 0, 1, \ldots, n-1,\ y_0 = u_a.$$

Геометрическая интерпретация метода состоит в замене точного решения ломаной (рис. 11.1). При этом угол наклона отрезка ломаной при $x \in [x_k, x_{k+1}]$ совпадает с углом наклона касательной к графику точного решения $u(x)$ в точке (x_k, u_k).

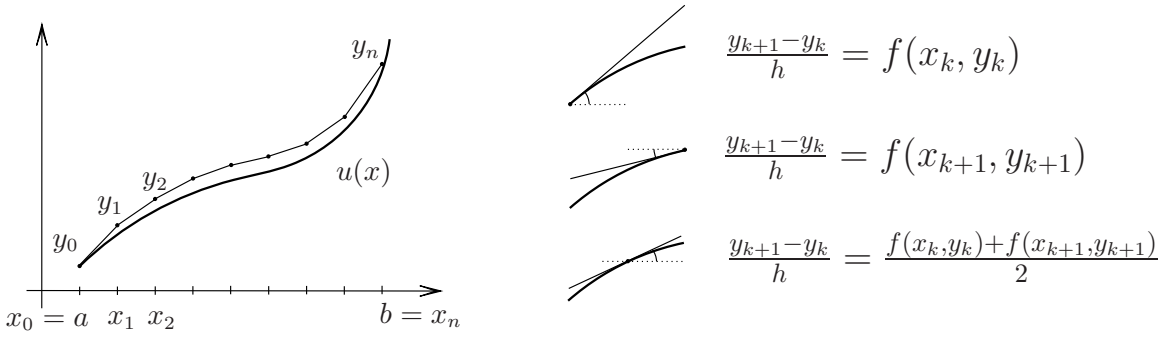


Рис. 11.1: Геометрическая интерпретация метода Эйлера. Справа три модификации метода для разных способов выбора углов наклона отрезков ломаной.

Найдём погрешность аппроксимации аппроксимации. Для этого нужно в разностное уравнение (11.5) подставить точное решение $u(x)$. Применим следующий метод. Зафиксируем узел x_k и выразим значение точного решения $u(x)$ в соседних узлах $x_{k\pm1}, x_{k\pm2}, \ldots$ в виде разложения в ряд Тейлора в окрестности точки x_k. В нашем случае в разностное выражение (11.5) входят только два соседних узла x_k и x_{k+1}. Представим $u_{k+1} = u(x_{k+1})$ в виде ряда Тейлора

$$u_{k+1} = u_k + hu'_{k+1} + O(h^2).$$

Подставим точное решение u_k и u_{k+1} дифференциального уравнения (11.2)

в разностное уравнение (11.5) вместо y_k и y_{k+1}. Вычитая из правой части левую, получим невязку

$$\psi_k = \frac{u_{k+1} - u_k}{h} - f(x_k, y_k) = u'_k + O(h) - f(x_k, y_k).$$

Из исходного уравнения (11.2) получаем, что $u'_k = f(x_k, y_k)$. Отсюда следует, что $\psi_n = O(h)$.

Интуитивно ясно, что качество приближения метода Эйлера определяется правильным выбором углов наклона отрезков ломаной. Можно рассмотреть модификацию метода с выбором угла наклона таким же, как у касательной к графику $u(x)$ в точке (x_{k+1}, u_{k+1}). Ясно, что в этом случае также будет $\psi_k = O(h)$.

Лучшее приближение будет достигнуто, если угол наклона отрезков ломанной будет заключен между углом касательной к $u(x)$ в точках (x_k, u_k) и (x_{k+1}, u_{k+1}). Так мы приходим к симметричной схеме.

Симметричная схема.

$$\frac{y_{k+1} - y_k}{h} = \frac{1}{2}(f(x_k, y_k) + f(x_{k+1}, y_{k+1})),\ k = 1, 2, \ldots n-1,\ y_0 = u_0. \quad (11.6)$$

Данный метод более сложен в реализации, чем метод Эйлера (11.5), так как новое значение y_{k+1} определяется по найденному ранее y_k путём решения уравнения

$$y_{k+1} - 0{,}5hf(x_{k+1}, y_{k+1}) = F_k,$$

где $F_k = y_n + 0{,}5hf(x_k, y_k)$. По этой причине метод называется *неявным*. Преимуществом метода (11.6) по сравнению с (11.5) является более высокий порядок точности.

Для невязки

$$\psi_k = \frac{u_{k+1} - u_k}{h} - \frac{1}{2}(u(x_k, y_k) + u(x_{k+1}, y_{k+1}))$$

справедливо разложение

$$\psi_k = u'_k + \frac{h}{2}u''_k + O(h^2) - \frac{1}{2}(u'_k + u''_{k+1}) = u'_k + \frac{h}{2}u''_k - \frac{1}{2}(u'_k + u'_k + hu''_k + O(h^2)),$$

т.е. $\psi_k = O(h^2)$. Таким образом, метод (11.6) имеет второй порядок аппроксимации, а, следовательно, и второй порядок точности.

Метод предиктор–корректор. Предположим, что приближённое значение y_k решения исходной задачи в точке $x = x_k$ уже известно. Для нахождения $y_{k+1} = y(x_{k+1})$ поступим следующим образом. Сначала, используя схему Эйлера

$$\frac{y_{k+1/2} - y_k}{0{,}5h} = f(x_k, y_k), \tag{11.7}$$

вычислим промежуточное значение $y_{k+1/2}$, а затем воспользуемся разностным уравнением

$$\frac{y_{k+1} - y_k}{h} = f(x_k + 0{,}5h, y_{k+1/2}), \tag{11.8}$$

из которого явным образом найдём искомое значение y_{k+1}.

Для исследования невязки подставим промежуточное значение $y_{k+1/2} = y_k + 0{,}5hf_k$, где $f_n = f(x_n, y_n)$, в уравнение (11.8). Тогда получим разностное уравнение

$$\frac{y_{k+1} - y_k}{h} = f(x_k + 0{,}5h, y_k + 0{,}5hf_k), \tag{11.9}$$

невязка которого равна

$$\psi_k = \frac{u_{k+1} - u_k}{h} - f(x_k + 0{,}5h, u_k + 0{,}5hf_k).$$

Имеем

$$\frac{u_{k+1} - u_k}{h} = u'_k + 0{,}5hu''_k + O(h^2).$$

Используя формулу Тейлора для функции нескольких переменных[8], получим окрестности точки (x_k, u_k)

$$\begin{aligned} f(x_k + 0{,}5h,\, u_k + 0{,}5hf(x_k, u_k)) &= f(x_k, u_k) + \\ &+ 0{,}5h\left(\frac{\partial f(x_k, u_k)}{\partial x} + f(x_k, u_k)\frac{\partial f(x_k, u_k)}{\partial u}\right) + O(h^2) = \\ &= f(x_k, u_k) + 0{,}5hu''_k + O(h^2), \end{aligned}$$

[8] **Теорема.** Пусть функция $u(x_1, x_2, \ldots, x_m)$ непрерывно дифференцируема $(n-1)$ раз в ε-окрестности точки $M_0(\mathring{x}_1, \mathring{x}_2, \ldots, \mathring{x}_m)$ и n раз дифференцируема в самой точке M_0. Тогда для любой точки M из указанной ε-окрестности M_0 справедлива следующая формула

$$u(M) = u(M_0) + \frac{1}{1!}du\Big|_{M_0} + \frac{1}{2!}d^2u\Big|_{M_0} + \ldots + \frac{1}{n!}d^nu\Big|_{M_0} + O(\rho^{n+1}),$$

где $\rho = \rho(M, M_0)$ — расстояние между точками M и M_0.

так как в силу (11.2) справедливо равенство $u' = f(x,u)$ и, следовательно,

$$u'' = \frac{\partial f}{\partial x} + f\frac{\partial f}{\partial u} \quad \text{(производная функции двух переменных)}.$$

Таким образом, метод (11.9) имеет второй порядок погрешности аппроксимации, $\psi_k = O(h^2)$, и в отличии от (11.6) является явным.

Реализация метода (11.9) в виде двух этапов (11.7), (11.8) называется *методом предиктор–корректор* (предсказывающе-исправляющим[9]), поскольку на первом этапе приближённое значение предсказывается с невысокой точностью $O(h)$, а на втором этапе это предсказанное значение исправляется, так что результирующая погрешность имеет второй порядок по h.

Тот же самый метод можно реализовать несколько иначе. А именно, сначала вычислим последовательно функции

$$p_1 = f(x_k, y_k), \quad p_2 = f(x_n + 0{,}5h,\ y_k + 0{,}5hp_1),$$

а затем найдём y_{k+1} из уравнения $(y_{k+1} - y_k)/h = p_2$.

Такая форма реализации метода предиктор–корректор называется методом Рунге–Кутта. Поскольку требуется вычислить две промежуточные функции p_1 и p_2, данный метод относится в *двухэтапным* методам.

Метод Рунге–Кутта.[10] *Явный m-этапный метод Рунге-Кутта* состоит в следующем. Пусть решение $y_k = y(x_k)$ уже известно. Задаются числовые коэффициенты

$$\begin{array}{ll} a_i, b_{ij}, & i = 2,3,\ldots,m,\ j = 1,2,\ldots,m-1, \\ \sigma_i, & i = 1,2,\ldots,m, \end{array}$$

и последовательно вычисляются функции

$$\begin{aligned} &p_1 = f(x_k, y_k), \\ &p_2 = f(x_k + a_2 h, y_k + b_{21} h p_1), \\ &p_3 = f(x_k + a_3 h, y_k + b_{31} h p_1 + b_{32} h p_2), \\ &\cdots \\ &p_m = f(x_k + a_m h, y_k + b_{m1} h p_1 + b_{m2} h p_2 + \ldots + b_{m,m-1} h p_{m-1}). \end{aligned}$$

Затем из формулы

$$\frac{y_{k+1} - y_k}{h} = \sum_{i=1}^{m} \sigma_i p_i \tag{11.10}$$

находится новое значение $y_{k+1} = y(x_{k+1})$.

[9] От англ. predict – предсказывать, correct – исправлять

[10] Ма́ртин Вильге́льм Ку́тта — немецкий физик и математик (1867–1944).

Коэффициенты a_i, b_{ij}, σ_i выбираются из соображений точности. Например, для того, чтобы уравнение (11.10) аппроксимировало исходное уравнение (11.2), необходимо потребовать $\sum\limits_{i=1}^{m} \sigma_i = 1$. Отметим, что методы Рунге–Кутта при $m > 5$ не используются.

Для получения очередного значения y_{k+1} в методе Рунге–Кутта требуется выполнить много промежуточных вычислений. Но из-за высокого порядка погрешности аппроксимации метода можно выбирать сетку с довольно крупным шагом h. В итоге общее число арифметических операций уменьшается. Наиболее распространённым является метод Рунге–Кутта четвёртого порядка:

$$\begin{cases} y_{k+1} = y_k + \frac{h}{6}(p_1 + 2p_2 + 2p_3 + p_4), \\ p_1 = f(x_k, y_k), \\ p_2 = f(x_k + \frac{h}{2}, y_k + \frac{h}{2}p_1), \\ p_3 = f(x_k + \frac{h}{2}, y_k + \frac{h}{2}p_2), \\ p_4 = f(x_k + h, y_k + hp_3). \end{cases}$$

Глава 2

Практические занятия

2.1 Занятие 1

2.1.1 Ошибки в вычислениях, числа с плавающей точкой

1.1 Пусть надо решить систему двух линейных уравнений:

$$-10^{-7}x_1 + x_2 = 1,$$
$$x_1 + 2x_2 = 4.$$

Решение. Первый метод. Исключая x_1 из первого уравнения: $x_1 = 10^7 x_2 - 10^7$, и подставляя это выражение во второе уравнение, получаем $x_2 = \frac{10^7+4}{10^2+2}$.

Проведя вычисления с семью значащими цифрами, получаем $x_2 = 1.000000$, $x_1 = 0.000000$, что совершенно неверно, как видно из второго уравнения.

Второй метод. Исключая x_1 из второго уравнения: $x_1 = 4 - 2x_2$, получаем для x_2 формулу $x_2 = \frac{1+4\cdot 10^{-7}}{1+2\cdot 10^{-7}}$. После вычислений получаем $x_2 = 1.000000$, $x_1 = 2.000000$ — правильное (с точностью по шести десятичных цифр) решение. □

Представление чисел с плавающей точкой в MATLAB По умолчанию переменные имеют тип «плавающая точка двойной точности». Каждое число занимает 8 байт. Представление имеет следующую структуру:

$$\overbrace{\underbrace{b_{63}}_{\text{знак } s}\quad\underbrace{b_{62}\ldots b_{52}}_{\text{порядок } e}\quad\underbrace{b_{51}\ldots b_0}_{\text{мантисса } f}}^{\text{8 байт}}$$

$$\text{число с пл. точкой} = \begin{cases} (-1)^s(2^{e-1023})(1.f) & \text{нормализованное, } 0 < e < 2047, \\ (-1)^s(2^{e-1022})(0.f) & \text{ненормализованное, } e = 0,\ f > 0, \\ \text{ошибка} & \text{иначе.} \end{cases}$$

Далеко не все действительные числа точно представимы числом с плавающей точкой. Множество F чисел с плавающей точкой имеет мощность $|F| \leqslant 2^{64}$, в то время как $|\mathbb{R}| = \infty$. Расстояние между числом $x \in F$ и ближайшим к нему числом $y \in F$ $(x < y)$ приблизительно пропорционально $|x|$. Чем дальше от нуля, тем реже встречаются числа из F среди чисел из $\mathbb{R}$.

Функция MATLAB`eps(x)` вычисляет расстояние до ближайшего числа с плавающей точкой, большего `x`.

1.2 Зная внутренне представление чисел системы MATLAB, определить, чему равно `eps(7)` и `eps(8)`. Ответы сравнить и объяснить, почему они отличаются в два раза.

Решение. Имеем $7_{10} = 111_2 = 1,11_2 \cdot 2^2 = (-1)^0(2^{1025-1023})(1,11)$. Отсюда $s = 0$, $e = 1025$, $f = 11\ \underbrace{00\ldots0}_{50\text{ нулей}}\ _2$. Следующее число $7'$ с плавающей точкой, большее 7, будет иметь $s = 0$, $e = 1025$, $f = 11\ \underbrace{00\ldots0}_{49\text{ нулей}}\ 1_2$. Получаем, что $eps(7) = 7' - 7 = 2^{-50} \approx 8.881784197001252 \cdot 10^{-16}$. Аналогично $eps(8) = 8' - 8 = 2^{-49} \approx 1.776356839400251 \cdot 10^{-15}$.

Ответ `eps(7)` в два раза меньше `eps(8)`. Так происходит всякий раз когда переходим через пороговое значение вида 2^k, $k \in \mathbb{Z}$. В нашем случае $7 < 2^3$, а $8 = 2^3$. □

1.3 Объяснить, почему при вычислении в среде MATLAB

```
100 + 2^-50
```

получается ответ 100, но

```
1 + 2^-50
```

возвращает ответ `1.000000000000001`?

Решение. Найдем двоичное представление первой суммы: $100 + 2^{-50} = 1100100_2 + 0,\ \underbrace{0\ldots0}_{49\text{ нулей}}\ 1_2 = 1100100,\ \underbrace{0\ldots0}_{49\text{ нулей}}\ 1_2$. Получилось 57 бит. Мантисса имеет длину 52 бита, то есть данное число необходимо округлить (отрезать «лишние» 5 бит). После округления остаётся только первое слагаемое 100.

Найдем двоичное представление второй суммы: $1+2^{-50} = 1_2+0,\ \underbrace{0\ldots0}_{49\text{ нулей}}\ 1_2 = 1,\ \underbrace{0\ldots0}_{49\text{ нулей}}\ 1_2$. Получилось 51 бит, что вполне умещается внутри мантиссы. □

Замечание. *52 двоичных разряда соответствуют 15–16 десятичным.*

2.1.2 Матричные вычисления в MATLAB

Полезные функции:

- `ones(m,n)` и `zeros(m,n)` — создать матрицу размера $m \times n$, все элементы которой 1 и 0. Пример:

$$\texttt{ones(2,2)} = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}, \qquad \texttt{zeros(1,3)} = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}.$$

- `eye(n)` — создать единичную матрицу размера $n \times n$. Пример:

$$\texttt{eye(3)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

- `diag(v,k)` — расположить элементы вектора v на k-ой диагонали квадратной матрицы соответствующего размера. Пример:

$$\texttt{v} = [1 \quad 2] \qquad \texttt{diag(v,0)} = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix},$$

$$\texttt{diag(v,1)} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 2 \\ 0 & 0 & 0 \end{pmatrix}, \qquad \texttt{diag(v,-1)} = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 2 & 0 \end{pmatrix}.$$

Пусть далее $A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$.

- `flipud(A)` (от англ. flip up down), `fliplr(A)` (от англ. flip left right), ' (штрих) — перевернуть матрицу сверху вниз, слева направо, относительно главной диагонали. Пример:

$$\texttt{flipud(A)} = \begin{pmatrix} 4 & 5 & 6 \\ 1 & 2 & 3 \end{pmatrix}, \quad \texttt{fliplr(A)} = \begin{pmatrix} 3 & 2 & 1 \\ 6 & 5 & 4 \end{pmatrix}, \quad \texttt{A'} = \begin{pmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{pmatrix}.$$

- `prod(A,k)`, `sum(x)` — перемножить, просуммировать элементы матрицы в направлении k. Пример:

$$\texttt{prod(A,1)} = \begin{pmatrix} 4 & 10 & 18 \end{pmatrix}, \qquad \texttt{prod(A,2)} = \begin{pmatrix} 6 \\ 120 \end{pmatrix},$$

$$\texttt{sum(A,1)} = \begin{pmatrix} 5 & 7 & 9 \end{pmatrix}, \qquad \texttt{sum(A,2)} = \begin{pmatrix} 6 \\ 15 \end{pmatrix}.$$

- `triu(A,k)` (от англ. upper triangle), `tril(A,k)` (от англ. lower triangle) — вернуть верхнюю, нижнюю треугольную часть матрицы A, начиная с k-ой диагонали. Пример:

$$\texttt{triu(A,1)} = \begin{pmatrix} 0 & 2 & 3 \\ 0 & 0 & 6 \end{pmatrix}, \qquad \texttt{tril(A,0)} = \begin{pmatrix} 1 & 0 & 0 \\ 4 & 5 & 0 \end{pmatrix}.$$

- `repmat(B,m,n)` — продублировать матрицу B по вертикали m раз и по горизонтали n раз. Пример:

$$B = \begin{pmatrix} 1 & 4 \end{pmatrix} \qquad \texttt{repmat(B,2,3)} = \begin{pmatrix} 1 & 4 & 1 & 4 & 1 & 4 \\ 1 & 4 & 1 & 4 & 1 & 4 \end{pmatrix}.$$

1.4 Пусть задан вектор `x = [1,2,4,5,6,7]`. Задать в MATLABматрицу определителя Вандермонда:

$$\Delta = \begin{pmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ \vdots & \vdots & & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{pmatrix}.$$

Решение. Неоптимальный подход:

```
x = [1,2,4,5,6,7];
Delta = zeros(length(x));      % зарезервировать память
for i=1:length(x)
    for j=1:length(x)
        Delta(i,j) = x(i)^(j-1);
    end
end
Delta          % вывести на экран ответ
```

Оптимальный подход:

```
x = [1,2,4,5,6,7];
temp = repmat(x', 1, length(x));
power = repmat(0:length(x)-1, length(x), 1);
Delta = temp.^power  % вычисляем и выводим на экран
```

Здесь имеем:

$$temp = \begin{pmatrix} x_0 & x_0 & x_0 & \cdots & x_0 \\ x_1 & x_1 & x_1 & \cdots & x_1 \\ \vdots & \vdots & & \ddots & \vdots \\ x_n & x_n & x_n & \cdots & x_n \end{pmatrix}, \qquad power = \begin{pmatrix} 0 & 1 & 2 & \cdots & n \\ 0 & 1 & 2 & \cdots & n \\ \vdots & \vdots & & \ddots & \vdots \\ 0 & 1 & 2 & \cdots & n \end{pmatrix}.$$

□

1.5 Пусть задан вектор `x = [1,2,4,5,6,7]`, `y = 2.5` и `k = 2`. Вычислить в MATLABфрагмент многочлена Лагранжа:

$$(y - x_0) \dots (y - x_{k-1})(y - x_{k+1}) \dots (y - x_n)$$

Решение. Неоптимальное:

```
x = [1,2,4,5,6,7];
k = 2;
y = 2.5;
result = 1;
for i=1:length(x)
    if i == (k+1)
        continue;
    end
    result = result * (y - x(i));
end
result    % выводим результат
```

Оптимальное решение:

```
x = [1,2,4,5,6,7];
k = 2;
y = 2.5;
result = y - x;
result(k+1) = 1;
result = prod(result)    % выводим результат
```

Здесь имеем последовательно:

$$result = [\ y - x_0\ \ \ldots\ \ y - x_{k-1}\ \ y - x_k\ \ y - x_{k+1}\ \ \ldots\ \ y - x_n\ ]$$
$$result = [\ y - x_0\ \ \ldots\ \ y - x_{k-1}\ \ 1\ \ y - x_{k+1}\ \ \ldots\ \ y - x_n\ ]$$
$$result = (y - x_0)\ldots(y - x_{k-1}) \cdot 1 \cdot (y - x_{k+1})\ldots(y - x_n).$$ □

1.6 Пусть задана переменная `m = 4`. Предложить оптимальное решение в MATLABдля построения следующей матрицы:

$$C = \begin{pmatrix} 0 & 1 & \ldots & m \\ 1 & 2 & \ldots & m+1 \\ 2 & 3 & \ldots & m+2 \\ \vdots & \vdots & \ddots & \\ m & m+1 & \ldots & m+m \end{pmatrix}$$

.

Указание. Представить искомую матрицу в виде $C = A + A^\top$. □

1.7 Пусть задан вектор `x = [1,2,4,5,6,7]` и `m = 4`. Вычислить в MATLABматрицу для метода среднеквадратического приближения:

$$\begin{pmatrix} n+1 & \sum x_i & \dots & \sum x_i^m \\ \sum x_i & \sum x_i^2 & \dots & \sum x_i^{m+1} \\ \sum x_i^2 & \sum x_i^3 & \dots & \sum x_i^{m+2} \\ \vdots & \vdots & \ddots & \\ \sum x_i^m & \sum x_i^{m+1} & \dots & \sum x_i^{m+m} \end{pmatrix}.$$

Решение. Заметим, что элементы матрицы принадлежат множеству $\{\sum x_i^0, \dots, \sum x_i^{2m}\}$. Вычислим соответствующий массив B:

$$A_{(2m+1\times n+1)} = \begin{pmatrix} x_0 & x_1 & \cdots & x_n \\ x_0 & x_1 & \cdots & x_n \\ \vdots & \vdots & & \vdots \\ x_0 & x_1 & \cdots & x_n \end{pmatrix} .\wedge \begin{pmatrix} 0 & \cdots & 0 \\ 1 & \cdots & 1 \\ \vdots & & \vdots \\ 2m & \cdots & 2m \end{pmatrix};$$

$$B = \texttt{sum(A,2)} = \begin{pmatrix} \sum x_i^0 & \sum x_i^1 & \cdots & \sum x_i^{2m} \end{pmatrix}^\top.$$

Ответом будет `D=C(B)`, где C — матрица из предыдущей задачи. Матрица D имеет тот же размер, что и C. Её элемент d_{ij} равен $b_{c_{ij}}$, т.е. элементу массива B с индексом c_{ij}. □

2.2 Занятие 2

2.2.1 Метод дихотомии.

2.1 Уравнение $f(x) = 0$ решают методом дихотомии. Известно, что $f(x)$ непрерывна на $[a, b]$ и $f(a)f(b) < 0$. Оцените сверху наибольшее число итераций, необходимое для получения корня с заданной точностью ε.

Решение. На каждой итерации исходный отрезок $[a, b]$ сужается в два раза: $b_n - a_n = (b-a)/2^n$. Критерий останова в методе дихотомии: $b_n - a_n < \varepsilon$.

Решая неравенство $(b-a)/2^n < \varepsilon$ и учитывая, что $n \in \mathbb{N}$, получим $n \geqslant \left[\log_2 \frac{b-a}{\varepsilon}\right] + 1$, где квадратные скобки — операция взятия *целой части* от числа. □

2.2.2 Метод простых итераций.

2.2 Методом простых итераций найти корни уравнения $x^2 - a = 0$ (квадратный корень из числа a).

Решение. 1 метод

$x = x^2 + x - a$, $\varphi(x) = x^2 + x - a$, $\varphi'(x) = 2x + 1$;

$|\varphi'(x)| < 1 \;\Rightarrow\; |2x+1| < 1 \;\Rightarrow\; \left|x + \frac{1}{2}\right| < \frac{1}{2}$

Только для $x \in (-1, 0)$ корень будет найден методом простых итераций.

2 метод

$x = \dfrac{a}{x}$, $\varphi(x) = \dfrac{a}{x}$, $\varphi'(x) = -\dfrac{a}{x^2}$.

Условие $|\varphi'(x) < 1|$, $\left|\frac{a}{x^2}\right| < 1$ или $|x| > \sqrt{a}$.

Но при $x_0 > \sqrt{a}$, $0 < x_1 < \dfrac{a}{x_0} < \sqrt{a}$ т.е. $0 < x_1 < \sqrt{a}$, и метод «зацикливается».

Например, $x_0 = 2\sqrt{a}$, $x_1 = \dfrac{1}{2}\sqrt{a}$, $x_2 = 2\sqrt{a}$, $x_3 = \dfrac{1}{2}\sqrt{a}$ и т.д.

3 метод

Запишем метод Ньютона

$$x = \varphi(x),\ \varphi(x) = x - \frac{f(x)}{f'(x)} = x - \frac{x^2 - a}{2x} = \frac{1}{2}\left(x + \frac{a}{x}\right). \qquad \square$$

2.3 Построить итерационный процесс вычисления корней уравнения $x^3 + 3x^2 - 1 = 0$ методом простой итерации.

Решение. Табличным способом выделим отрезки, на концах которых функция $f(x)$ имеет разные знаки

x	-3	-2	-1	0	1	2	3
$\operatorname{sign} f(x)$	$-$	$+$	$+$	$-$	$+$	$+$	$+$

Таким образом, корни исходного уравнения лежат на отрезках $[-3, -2]$, $[-1, 0]$ и $[0, 1]$, для каждого из которых построим свой итерационный процесс.

Для $x \in [-3,-2]$ разделим исходное уравнение на x^2. В результате получим равносильное уравнение $x = \varphi(x)$, $\varphi(x) = \frac{1}{x^2} - 3$. Итерационный процесс для нахождения первого корня: $x_{n+1} = \frac{1}{x_n^2} - 3$. Поскольку $|\varphi'(x)| = \left|-\frac{2}{x^3}\right| \leqslant \frac{1}{4} < 1$ для $x \in [-3,-2]$, то сходимость имеет место для всех начальных приближений $x_0 \in [-3,-2]$.

Для двух других отрезков исходное уравнение перепишем в виде $x^2(x+3) - 1 = 0$. Если $x_0 \in [-1,0]$, то определим итерационный процесс $x_{n+1} = -\frac{1}{\sqrt{x_n+3}}$; если $x_0 \in [0,1]$, то $x_{n+1} = \frac{1}{\sqrt{x_n+3}}$. Можно показать, что в процессе итераций соответствующие отрезки отображаются в себя, поэтому сходимость построенных итерационных процессов следует из оценки $|\varphi'(x)| = \frac{1}{2}\left|\frac{1}{\sqrt{x+3}}\right|^3 < 1$. □

2.4 Пусть $\varphi(x)$ непрерывно дифференцируема и $\varphi'(x)$ не меняет свой знак на интервале $[x_n, x_{n+1}]$. Показать, что если $\varphi'(x) > 0$, то x_n и x_{n+1} лежат по одну сторону от корня x_*; если же $\varphi'(x) < 0$, то корень x_* заключен между соседними элементами x_n и x_{n+1}.

Решение. По теореме Лагранжа $x_{n+1} - x_* = \varphi(x_n) - \varphi(x_*) = (x_n - x_*)\varphi'(\xi)$, где точка ξ лежит между точками x_n, x_*. Если $\varphi'(x) > 0$, то разности $(x_{n+1} - x_*)$ и $(x_n - x_*)$ должны иметь одинаковый знак; если $\varphi'(x) < 0$, то знаки разностей должны отличаться. □

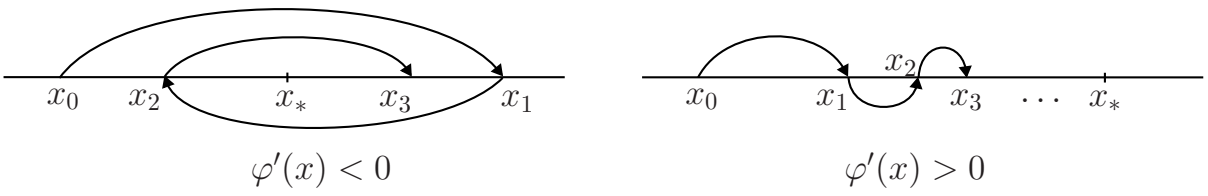


Рис. 2.1: Задача 2.4

2.5 Определить область начальных приближений x_0, для которых итерационный процесс $x_{n+1} = \dfrac{x_n^3 + 1}{20}$ сходится.

Решение. Уравнение $x^3 - 20x + 1 = 0$ имеет три различных вещественных корня: $z_1 < z_2 < z_3$. В зависимости от выбора начального приближения x_0 итерационный процесс либо расходится, либо сойдется к одному из корней z_i, $i = 1, 2, 3$.

Перепишем формулу итерационного процесса в виде

$$x_{n+1} - x_n = \frac{x_n^3 - 20x_n + 1}{20} = \frac{(x_n - z_1)(x_n - z_2)(x_n - z_3)}{20}.$$

Если $x_n < z_1$, то $x_{n+1} - x_n < 0$ и последовательность x_n монотонно убывает. Это означает расходимость итерационного процесса при $x_0 < z_1$, так как $x_n < x_0 < z_i$, $i = 1, 2, 3$. Аналогично показывается, что при $z_3 < x_0$ выполняются неравенства $z_i < x_n < x_{n+1}$, и метод расходится.

Точки $x_0 = z_1$, $x_0 = z_2$ и $x_0 = z_3$ являются неподвижными, а отображение $x_{n+1} = (x_n^3 + 1)/20$ монотонно. Поэтому для $z_1 < x_0 < z_2$ имеем $z_1 < x_n < x_{n+1} < z_2$. Таким образом, последовательность x_n монотонно возрастает, ограничена сверху и сходится к точке z_2. Аналогично доказывается, что для $x_0 \in (z_2, z_3)$ последовательность x_n, монотонно убывая, сходится к z_2. □

2.6 Уравнение $x + \ln x = 0$, имеющее корень $z \approx 0,6$, предлагается решить одним из методов простой итерации:

1) $x_{n+1} = -\ln x_n$; 2) $x_{n+1} = e^{-x_n}$;

3) $x_{n+1} = \dfrac{x_n + e^{-x_n}}{2}$; 4) $x_{n+1} = \dfrac{3x_n + 5e^{-x_n}}{8}$.

Исследовать эти методы и сделать выводы о целесообразности использования каждого из них.

2.7 Уравнение $x = 2^{x-1}$, имеющее два корня $z_1 = 1$ и $z_2 = 2$, решается методом простой итерации. Исследовать его сходимость в зависимости от выбора начального приближения x_0.

2.8 Найти область сходимости метода простой итерации для следующих уравнений:

1) $x = e^{2x} - 1$, 2) $x = 1/2 - \ln x$, 3) $x = \operatorname{tg} x$.

2.9 Доказать, что итерационный процесс $x_{n+1} = \cos x_n$ сходится для любого начального приближения $x_0 \in \mathbb{R}$.

Решение. При любом $x_0 \in \mathbb{R}$ $x_1 \in [-1, 1]$ и вообще $x_n \in [-1, 1]$, $n \geqslant 1$. Имеем $\varphi(x) = \cos(x)$, $\varphi'(x) = -\sin(x)$. Получаем, что $|\varphi'(x)| < 1$ при $x \in [-1, 1]$. Функция $\varphi(x)$ удовлетворяет условию Коши-Липшица, поэтому итерационный процесс сходится. □

2.2.3 Метод Ньютона.

2.10 Построить итерационный процесс Ньютона для вычисления $\sqrt[p]{a}$, $a > 0$, где $p \in \mathbb{R}$.

Значение $\sqrt[p]{a}$ является корнем уравнения $f(x) = x^p - a = 0$. Для этого уравнения метод Ньютона имеет вид

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} = x_n - \frac{x_n^p - a}{p x_n^{p-1}} = \frac{p-1}{p} x_n + \frac{a}{p x_n^{p-1}}.$$

Для $p = 2$ получаем $x_{n+1} = \dfrac{1}{2}\left(x_n + \dfrac{a}{x_n}\right)$.

Говорят, что z является корнем кратности p, если

$$f(z) - f'(z) = \ldots = f^{(p-1)}(z) = 0, \qquad f^{(p)}(z) \neq 0.$$

Это равносильно следующему:
Если функция представима в виде $f(p) = (x - z)^p g(x)$, $p \in \mathbb{N}$, а в некоторой окрестности точки z выполняется $|g(x)| < \infty$, $g(z) \neq 0$, то p называют кратностью корня.

Для уравнения $f(x) = 0$ формула метода Ньютона имеет вид

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

для нахождения простых корней и

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

для нахождения корней кратности p.

2.11 Определить кратность корня $z = 2$ для уравнения

$$x^3 - 7x^2 + 16x - 12 = 0.$$

2.3 Занятие 3

2.3.1 Интерполяция по Лагранжу и Ньютону. Оценка остаточного члена.

3.1 Построить многочлен Лагранжа при $n = 3$ для следующих случаев:

1) $x_1 = -1, \ x_2 = 0, \ x_3 = 1,$ $f_1 = 3, \quad f_2 = 2, \ f_3 = 5;$ 2) $x_1 = 1, \ x_2 = 2, \ x_3 = 4,$ $f_1 = 3, \ f_2 = 4, \ f_3 = 6;$

Решение.

$$1) \quad P_3(x) = f_1\frac{(x-x_2)(x-x_3)}{(x_1-x_2)(x_1-x_3)} + f_2\frac{(x-x_1)(x-x_3)}{(x_2-x_1)(x_2-x_3)} + \\ + f_2\frac{(x-x_1)(x-x_2)}{(x_3-x_1)(x_3-x_2)} = 3\frac{(x-0)(x-1)}{(-1-0)(-1-1)} + \\ + 2\frac{(x-(-1))(x-1)}{(0-(-1))(0-1)} + 5\frac{(x-(-1))(x-0)}{(1-(-1))(1-0)} = 2x^2 + x + 2.$$

□

3.2 Приближение к числу $\ln 15{,}2$ вычислено следующим образом. Найдены точные значения $\ln 15$ и $\ln 16$ и построена линейная интерполяция между этими числами. Показать, что если a и a^* — соответственно точное и интерполированное значения $\ln 15{,}2$, то справедлива оценка $0 < a - a^* < 4 \cdot 10^{-4}$.

Решение. Запишем погрешность $R(x) \leqslant \frac{|M_2|}{2!}|(x-15)(x-16)|$, где $M_2 = \max\limits_{15\leqslant x\leqslant 16} |(\ln x)''| = \max\limits_{15\leqslant x\leqslant 16} 1/x^2 = 1/225$. Нас интересует погрешность в конкретной точке: $R(15{,}2) \leqslant \dfrac{0,2 \cdot 0,8}{225 \cdot 2} < 3 \cdot 10^{-4}$. □

3.3 Построить интерполяционный многочлен для функции $f(x) = |x|$ по узлам $-1, 0, 1$.

3.4 Построить интерполяционный многочлен для функции $f(x) = x^2$ по узлам $0, 1, 2, 3$.

3.5 С каким шагом следует составлять таблицу функции $\sin x$ на отрезке $[0, \pi/2]$, чтобы погрешность кусочно-линейной интерполяции не превосходила величины $0,5 \cdot 10^{-6}$?

3.6 Построить многочлен $P_3(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3$, удовлетворяющий условиям:

1. $P_3(-1) = 0$, $P_3(1) = 1$, $P_3(2) = 2$, $a_3 = 1$.
2. $P_3(0) = P_3(-1) = P_3(1) = 0$, $a_2 = 1$.
3. $P_3(-1) = 0$, $P_3(1) = 1$, $P_3(2) = 2$, $a_1 = 1$.
4. $P_3(-2) = P_3(-1) = P_3(1) = 0$, $a_0 = 1$.

2.3.2 Многочлены Чебышева

3.7 Вычислить многочлен Чебышёва $T_6(x)$ с помощью рекуррентного соотношения:
$T_0(x) = 1, \quad T_1(x) = x, \quad T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x), \quad n \geqslant 1.$

Решение.

$$
\begin{aligned}
T_2(x) &= 2T_1(x) - T_0(x) = 2x^2 - 1,\\
T_3(x) &= 2T_2(x) - T_1(x) = 4x^3 - 3x,\\
T_4(x) &= 2T_3(x) - T_2(x) = 8x^4 - 8x^2 + 1,\\
T_5(x) &= 2T_4(x) - T_3(x) = 16x^5 - 20x^3 + 5x,\\
T_6(x) &= 2T_5(x) - T_4(x) = 32x^6 - 48x^4 + 18x^2 - 1.
\end{aligned}
$$

□

3.8 Найти все нули многочлена Чебышёва $T_n(x)$.

Решение. Воспользуемся тригонометрической формой записи многочлена Чебышёва: $T_n(x) = \cos(n \cdot \arccos x)$. Решая уравнение $\cos(n \cdot \arccos x) = 0$, получим $x_k^{(n)} = \cos \frac{2k-1}{2n}\,\pi, \qquad k = 1, 2, \ldots, n.$ □

3.9 Найти многочлен, наименее уклоняющийся от нуля на отрезке $[a, b]$, среди всех многочленов со старшим коэффициентом 1.

Решение. Функция может $T_n(x)$ принимает аргумент из интервала $[-1, 1]$, а нам нужно подавать аргумент из интервала $[a, b]$. Найдём линейное преобразование $[a, b] \to [-1, 1]$. Легко заметить, что преобразование $x' = \frac{2x-(b+2)}{b-a}$ обладает нужным свойством. Сначала обратим внимание на главный коэффициент в представлении $T_n(x) = 2^{n-1}x^n + \ldots$. Он равен 2^{n-1}. Теперь рассмотрим главный коэффициент при подстановке x':

$$T_n(x') = T_n\left(\frac{2x-(b+2)}{b-a}\right) = 2^{n-1}\left(\frac{2x-(b+2)}{b-a}\right)^n + \ldots = \frac{2^{2n-1}}{(b-a)^n}x^n + \ldots$$

Умножая весь многочлен на коэффициент $(b-a)^n 2^{1-2n}$, добъёмся, чтобы главный множитель стал равен 1:

$$\overline{T}_n^{[a,b]}(x) = (b-a)^n 2^{1-2n} T_n\left(\frac{2x-(b+2)}{b-a}\right) = 1 \cdot x^n + \ldots.$$

Осталось показать, что $\overline{T}_n^{[a,b]}(x)$ — наименее уклоняющийся от нуля на отрезке $[a, b]$, среди всех многочленов со старшим коэффициентом 1. Доказательство аналогично случаю $\overline{T}_n^{[-1,1]}(x)$ (см. лекцию 4). □

3.10 Среди всех многочленов вида $a_3x^3 + 2x^2 + a_1x + a_0$ найти наименее уклоняющийся от нуля на отрезке $[3, 5]$ (т.е. многочлен вида $\mathrm{const} \cdot T_3^{[3,5]}(x)$).

Важное замечание. В задаче зафиксирован не главный коэффициент при x^3, а множитель при x^2. Теорема *о наименее отклоняющемся от нуля многочлене* не работает в этом случае. То есть для всех многочленов $P_3(x)$ степени 3 с одинаковым коэффициентом 2 при x^2 неравенство $\max\limits_{[3,5]} |P_3(x)| \geqslant \max\limits_{[3,5]} |\widetilde{T}_3^{[3,5]}(x)|$, где $\widetilde{T}_3^{[3,5]}(x)$ имеет коэффициент

2 при x^2 и корни как у $T_3^{[3,5]}(x)$, в общем случае не выполняется. Но термин «наименее отклоняющийся от нуля» нужно понимать как синоним «многочлен чебышёвского типа», т.е. многочлен с корнями как у $T_3^{[3,5]}(x)$.

Решение. Построим вначале многочлен Чебышёва $T_3^{[-1,1]}(x) = 4x^3 - 3x$. В условии задан отрезок $[3,5]$. Линейное преобразование $\frac{2x-(5+3)}{5-3} = x - 4$ осуществляет перевод $[3,5] \to [-1,1]$. Получаем $T_3^{[3,5]}(x) = T_3^{[-1,1]}(x-4) = 4x^3 - 48x^2 + 189x - 244$. Нам задан коэффициент при x^2. Это значит, что многочлен нужно разделить на -24: $-\frac{x^3}{6} + 2x^2 - \frac{63}{8} + \frac{61}{6}$. □

3.11 Функция $f(x) = \sin 2x$ приближается многочленом Лагранжа на $[0,2]$ по n чебышёвским узлам: $x_i = 1 + \cos\frac{2i-1}{2n}\pi$, $i = 1, \ldots, n$. Найти наибольшее целое p в оценке погрешности вида $\varepsilon_n = \frac{1}{3}10^{-p}$, если $n = 6$.

Решение. Для интерполяции используется 6 чебышёвских узлов, следовательно, степень многочлена Лагранжа на единицу меньше: $P_5(x)$. Погрешность многочлена Лагранжа:

$$R_5(x) = f(x) - P_5(x) = \frac{(\sin x)^{\mathrm{VI}}|_{x=\xi}}{(5+1)!} \cdot \omega_6(x),$$

где $\xi \in [0,2]$ — неизвестная точка из интервала и

$$\omega_6(x) = (x - x_1)(x - x_2)\ldots(x - x_6), \quad x_i \text{ — чебышёвские узлы.}$$

Оценим сверху модуль производной $M_6 = \max\limits_{[0,2]} |(\sin x)^{\mathrm{VI}}| = 2^6$. Имеем $\omega_6(x) = \overline{T}_6^{[0,2]}(x)$ — многочлен чебышёвского типа, у которого все 6 корней принадлежат $[0,2]$ и старший коэффициент равен 1. Найдём $\max\limits_{[0,2]} |\omega_6(x)| = \max\limits_{[0,2]} |\overline{T}_6^{[0,2]}(x)| \overset{\text{см. лекц. 4}}{=} (2-0)^6 \cdot 2^{1-2\cdot 6} = 2^{-5}$. В итоге

$$|R_6(x)| \leqslant \frac{M_6}{6!} \cdot \max_{[0,2]} |\omega_6(x)| = \frac{2^6}{720} \cdot 2^{-5} = \frac{1}{360} \lessapprox \frac{1}{3} \cdot 10^{-2}.$$

Ответ: $p = 2$. □

2.4 Занятие 4

2.4.1 Среднеквадратическое приближение

4.1 Провели эксперимент по измерению координаты x движущегося прямолинейно тела в зависимости от времени t. Были получены следующие данные:

t	0	1	2	3
x	2	7,1	13,9	23,2

. Получить с помощью метода наименьших квадратов зависимость $x(t)$, если известно, что тело движется равноускоренно (координата x имеет квадратичную зависимость от времени t).

Решение. Координата тела, движущегося прямолинейно под действием постоянного ускорения выражается квадратичной зависимостью:

$$x(t) = a + bt + ct^2.$$

С помощью метода наименьших квадратов определим числа a, b, c. Для каждого значения t найдём квадрат невязки и затем все невязки сложим:

$$\Phi(a,b,c) = (\underbrace{a-2}_{t=0})^2 + (\underbrace{a+b+c-7{,}1}_{t=1})^2 + (\underbrace{a+2b+4c-13{,}9}_{t=2})^2 + \\ + (\underbrace{a+3b+9c-23{,}2}_{t=3})^2.$$

Найдём минимум $\Phi(a,b,c)$:

$$\frac{\partial\Phi}{\partial a} = 2(a-2)+2(a+b+c-7{,}1)+2(a+2b+4c-13{,}9)+2(a+3b+9c-23{,}2) = \\ = 8a + 12b + 28c - 92{,}4 = 0$$

$$\frac{\partial\Phi}{\partial b} = 2(a+b+c-7{,}1) + 4(a+2b+4c-13{,}9) + 6(a+3b+9c-23{,}2) = \\ = 12a + 28b + 72c - 209 = 0$$

$$\frac{\partial\Phi}{\partial c} = 2(a+b+c-7{,}1) + 8(a+2b+4c-13{,}9) + 18(a+3b+9c-23{,}2) = \\ = 28a + 72b + 196c - 543 = 0$$

Решая совместно последние три уравнения получим: $a = 2{,}04$; $b = 3{,}89$; $c = 1{,}05$. Окончательно $x(t) = 2{,}04 + 3{,}89t + 1{,}05t^2$. □

4.2 Определить среднеквадратическое отклонение для результата предыдущей задачи.

Решение. Для каждого момента времени определим невязку по формуле $\delta_i = x(t_i) - x_i$, $i = 0, 1, \ldots, 3$ ($x(t_i)$ — зависимость, полученная методом наименьших квадратов; x_i — значение координаты из таблицы):

t	0	1	2	3
x	2	7,1	13,9	23,2
δ	0,04	−0,12	0,12	−0,14

Вычислим среднеквадратическое отклонение

$$\sqrt{\frac{1}{4}(\delta_0^2 + \delta_1^2 + \delta_2^2 + \delta_3^2)} = \sqrt{\frac{1}{4}(0,04^2 + 0,12^2 + 0,12^2 + 0,14^2)} \approx 0,11.$$

□

4.3 Найти обобщённое решение (в смысле метода наименьших квадратов) переопределённой системы

$$\begin{cases} x + y = 1, \\ x - y = 2, \\ 2x + y = 2,4. \end{cases}$$

Решение. Составим $\Phi(x, y) = (x + y - 1)^2 + (x - y - 2)^2 + (2x + y - 2,4)^2$.

$$\frac{\partial \Phi}{\partial x} = 2(x + y - 1) + 2(x - y - 2) + 4(2x + y - 2,4) = 12x + 4y = 15,6 = 0,$$

$$\frac{\partial \Phi}{\partial y} = 2(x + y - 1) + 2(x - y - 2) + 2(2x + y - 2,4) = 8x + 2y = 10,8 = 0.$$

При решении системы из двух уравнений, получим: $x = 1,5$; $y = -0,6$. Невязки: $|\delta_1| = 0,1$; $|\delta_2| = 0,1$; $|\delta_3| = 0$. □

2.4.2 Численное дифференцирование

4.4 Написать с помощью метода неопределённых коэффициентов формулу для вычисления y_1'' по равноотстоящим узлам x_0, x_1, x_2, x_3.

Решение. Метод неопределённых коэффициентов даёт результат, аналогичный разложению $y(x)$ в ряд Тейлора и выделению слагаемого y_1''. Если у нас 4 узла, то в разложении функции $y(x)$ в ряд Тейлора у нас будут участвовать четыре слагаемых (столько же, сколько узлов), содержащие: $y(x)$, $y'(x)$, $y''(x)$ и $y'''(x)$. То есть y_1'' будет выражаться через $y(x)$, $y'(x)$, $y''(x)$ и $y'''(x)$, а погрешность будет выражаться через оставшиеся члены: $y^{IV}(x), y^{V}(x), \ldots$ Ясно, что такая формула будет точна, если $y^{IV}(x) = y^{V}(x) = \ldots = 0$. Последнее будет справедливо для любого многочлена степени не выше 3. В частности, рассмотрим следующие четыре многочлена (столько же, сколько узлов)

$$z = 1, \quad z = x - x_0, \quad z = (x - x_0)^2, \quad z = (x - x_0)^3$$

(выбор именно этих многочленов обусловлен удобством дальнейших расчётов).

Несложно показать, что, если подобрать числа c_k таким образом, чтобы $z_1'' = \sum\limits_{k=0}^{3} z_k c_k$ было точным равенством (для выбранных четырёх многочленов z), то это будет значение $c_k(x)$ для многочлена Лагранжа в точке $x = x_1$.

Найдём производные:

z	1	$x - x_0$	$(x - x_0)^2$	$(x - x_0)^3$
z''	0	0	2	$6(x - x_0)$

Для каждой колонки запишем $z_1'' = \sum\limits_{k=0}^{3} z_k c_k$:

$$\begin{aligned}
0 &= c_0 \cdot 1 + c_1 \cdot 1 + c_2 \cdot 1 + c_3 \cdot 1, \\
0 &= c_0(x_0 - x_0) + c_1(x_1 - x_0) + c_2(x_2 - x_0) + c_3(x_3 - x_0), \\
2 &= c_0(x_0 - x_0)^2 + c_1(x_1 - x_0)^2 + c_2(x_2 - x_0)^2 + c_3(x_3 - x_0)^2, \\
6(x_1 - x_0) &= c_0(x_0 - x_0)^3 + c_1(x_1 - x_0)^3 + c_2(x_2 - x_0)^3 + c_3(x_3 - x_0)^3.
\end{aligned}$$

Узлы x_i, $i = 0, 1, 2, 3$ — равноотстоящие с шагом h, поэтому $x_1 - x_0 = h$,

$x_2 - x_0 = 2h$, $x_3 - x_0 = 3h$:

$$\begin{aligned} 0 &= c_0 + c_1 + c_2 + c_3, \\ 0 &= c_1 + 2c_2 + 3c_3, \\ 2 &= c_1 h^2 + 4c_2 h^2 + 9c_3 h^2, \\ 6 &= c_1 h^2 + 8c_2 h^2 + 27c_3 h^2. \end{aligned}$$

Решая систему, получим:

$$c_0 = \frac{1}{h^2}, \quad c_1 = -\frac{2}{h^2}, \quad c_2 = \frac{1}{h^2}, \quad c_3 = 0.$$

Окончательно $y_1'' \approx \frac{1}{h^2}(y_0 - 2y_1 + y_2)$. □

4.5 С помощью разложения в ряд Тейлора определить порядок погрешности в формуле из предыдущей задачи.

Решение. На основании решения предыдущей задачи имеем:

$$y_1'' = \frac{1}{h^2}(y_0 - 2y_1 + y_2) + O(h^p),$$

где порядок p нужно определить. Для этого разложим y_i, $i = 0, 2$ в ряд Тейлора в окрестности точки x_1 до члена h^4 (сейчас точно неясно до какого члена нужно раскладывать, но, если понадобится, разложение всегда можно продолжить):

$$\begin{aligned} y(x_0) = y(x_1) + (x_0 - x_1)y'(x_1) + \frac{(x_0 - x_1)^2}{2!}y''(x_1) + \\ + \frac{(x_0 - x_1)^3}{3!}y'''(x_1) + \frac{(x_0 - x_1)^4}{4!}y^{\mathrm{IV}}(x_1) + O((x_0 - x_1)^5), \end{aligned}$$

$$\begin{aligned} y(x_2) = y(x_1) + (x_2 - x_1)y'(x_1) + \frac{(x_2 - x_1)^2}{2!}y''(x_1) + \\ + \frac{(x_2 - x_1)^3}{3!}y'''(x_1) + \frac{(x_2 - x_1)^4}{4!}y^{\mathrm{IV}}(x_1) + O((x_2 - x_1)^5), \end{aligned}$$

В более компактной записи:

$$y_0 = y_1 - hy_1' + h^2 y_1''/2 - h^3 y_1'''/6 + h^4 y_1^{\mathrm{IV}}/24 + O(h^5),$$

$$y_2 = y_1 + hy_1' + h^2 y_1''/2 + h^3 y_1'''/6 + h^4 y_1^{\mathrm{IV}}/24 + O(h^5).$$

Подставим полученные значения в формулу для численной производной:

$$\frac{1}{h^2}(y_0 - 2y_1 + y_2) = \frac{1}{h^2}\left[h^2 y_1'' + \frac{h^4}{12} y_1^{\mathrm{IV}} + O(h^5)\right] = y_1'' + \underbrace{\frac{h^2}{12} y_1^{\mathrm{IV}} + O(h^3)}_{O(h^2)}.$$

Так как четвёртая производная в общем случае отлична от нуля, то порядок погрешности равен 2. □

2.5 Занятие 5

2.5.1 Численное дифференцирование (продолжение)

5.1 Найти оптимальное значение шага при вычислении $(\sqrt{x})'|_{x=105}$ на компьютере, используя числа с плавающей точкой двойной точности, по формуле центральной разности с машинной точностью. Определить с какой точностью будет вычислена производная.

Решение. Из-за округления чисел в мантиссе любые вычисления на компьютере ограничены машинной точностью δ. Формула *средней разностной производной* с шагом h:

$$y'(x) \approx \frac{f(x+h) - f(x-h)}{2h}.$$

В действительности компьютер вычисляет

$$y'(x) \approx \frac{\widetilde{y}(x+h) - \widetilde{y}(x-h)}{2h}, \text{ где } \widetilde{y} = y \pm \delta.$$

Для краткости пусть $y_+ = y(x+h)$, $y_- = y(x-h)$. Рассмотрим погрешность

$$\begin{aligned}\Delta = \left|y' - \frac{\tilde{y}_+ - \tilde{y}_-}{2h}\right| &= \left|\left(y' - \frac{y_+ - y_-}{2h}\right) + \left(\frac{\tilde{y}_+ - y_+}{2h} - \frac{\tilde{y}_- - y_-}{2h}\right)\right| \leqslant \\ &\leqslant \left|y' - \frac{y_+ - y_-}{2h}\right| + \left|\frac{\tilde{y}_+ - y_+}{2h}\right| + \left|\frac{\tilde{y}_- - y_-}{2h}\right| \leqslant \\ &\leqslant \underbrace{\frac{|y'''(\eta)|}{3} h^2}_{\substack{\text{см. разложение} \\ \text{в лекции №5}}} + \frac{\delta}{2h} + \frac{\delta}{2h} \leqslant \frac{M_3}{3} h^2 + \frac{\delta}{h} = \Phi(h),\end{aligned}$$

где $M_3 = \max\limits_{[x-h,x+h]} |y'''(x)|$.

Минимизируем ошибку $\Phi(h)$:

$$h_{\text{opt}}: \quad \Phi'(x) = \frac{2M_3 h}{3} - \frac{\delta}{h^2} = 0 \quad \Rightarrow \quad h_{\text{opt}} = \sqrt[3]{\frac{3\delta}{2M_3}}.$$

Остаётся определить значения δ и M_3.

Найдём погрешность входных данных δ. Порядок $\sqrt{x}$, где $x \sim 105$ равен ~ 10 Будем считать, что при использовании чисел с плавающей точкой двойной точности машинная погрешность составляет 16 десятичных разрядов. Из представления

$$\sqrt{105} = \underbrace{10{,}x\ldots\ldots x}_{\text{16 десят. разр.}} \,|\, \underbrace{xx\ldots\ldots}_{\substack{\text{не поместилось} \\ \text{в мантиссе}}}$$

машинная точность в нашей задаче составит $\delta = \pm 10^{-14}$.

Вычислим теперь M_3. Найдем производные для нашей задачи:

$$y' = \frac{1}{2\sqrt{x}}, \quad y'' = -\frac{1}{4x\sqrt{x}}, \quad y''' = \frac{3}{8x^2\sqrt{x}}.$$

$y'''(x)$ монотонно убывает при $x > 0$. Следовательно в качестве верхней оценки $M_3 = \max\limits_{[105-h,105+h]} y'''(x)$ можно взять легко вычисляемое значение $y'''(100) = \frac{3}{8\cdot 10^5} = 3{,}75 \cdot 10^{-6}$.

$$h_{\text{opt}} = \sqrt[3]{\frac{3}{2} \cdot \frac{10^{-14}}{3{,}75 \cdot 10^{-6}}} = \sqrt[3]{4 \cdot 10^{-9}} \approx 1{,}6 \cdot 10^{-3}.$$

$$\Phi(h_{\text{opt}}) \approx \frac{3{,}75 \cdot 10^{-6} \cdot (1{,}6 \cdot 10^{-3})^2}{3} + \frac{10^{-14}}{1{,}6 \cdot 10^{-3}} = 3{,}2 \cdot 10^{-12} + 6 \cdot 10^{-12} \sim$$

$$\sim 10^{-11} \gg 10^{-14} = \delta.$$

□

Обратим внимание, что погрешность численной производной на 3 порядка превосходит погрешность входных данных.

5.2 Методом Рунге уточнить правую разностную производную и получить в итоге формулу с погрешностью третьего порядка.

2.5.2 Численное интегрирование

5.3 Доказать, что $\int_a^b f(x)\,dx = \sum_{i=1}^n f(x_{i-1})h + O(h)$ (формула левых прямоугольников).

Решение. На лекции была рассмотрена формула центральных прямоугольников $\int_a^b f(x)\,dx = \sum_{i=1}^n f(x_{i-1/2})h + O(h^2)$. Её порядок погрешности равен 2. В условии задачи приведена формула с меньшим порядком погрешности.Перейдём к доказательству.

Общая погрешность равна

$$\Psi = \int_a^b f(x)\,dx - \sum_{i=1}^n f(x_{i-1})h = \sum_{i=1}^n \underbrace{\int_{x_{i-1}}^{x_i} [f(x) - f(x_{i-1})]\,dx}_{\psi_i}.$$

Подставим погрешность многочлена Лагранжа

$$\psi_i = \int_{x_{i-1}}^{x_i} \frac{f'(x)}{1!}(x - x_{i-1})\,dx,$$

$$|\psi_i| \leqslant M_{1,i} \int_{x_{i-1}}^{x_i} (x - x_{i-1})\,dx = \frac{M_{1,i}}{2}h^2, \quad \text{где } M_{1,i} = \max_{[x_{i-1},x_i]} |f'(x)|.$$

$$|\Psi| = \left|\sum_{i=1}^n \psi_i\right| \leqslant \sum_{i=1}^n |\psi_i| = \sum_{i=1}^n \frac{M_{1,i}}{2}h^2 \leqslant \frac{M_1 h}{2} \underbrace{\sum_{i=1}^n h}_{b-a} = \frac{M_1(b-a)}{2}h = O(h),$$

где $M_1 = \max\limits_{[a,b]} |f'(x)|$. □

5.4 Показать, что $\int\limits_{-2h}^{2h} |\omega_5(x)|\,dx = \frac{19}{3}h^6$.

5.5 Вычислить интеграл $\int\limits_0^1 \exp(x^2)\,dx$ по формуле Ньютона–Котеса с узлами 0, 1/4, 1/2, 3/4, 1 и оценить погрешность.

Решение. Задано 5 равноотстоящих узлов. Приблизим функцию многочленом Лагранжа $\exp(x^2) \approx P_4(x)$. Отсюда

$$\int_0^1 \exp(x^2)\,dx \approx \int_0^1 P_4(x)\,dx = \int_0^1 \sum_{k=0}^{4} \exp(x_k^2)\frac{L_4^{(k)}(x)}{L_4^{(k)}(x_k)}\,dx =$$

$$= \sum_{k=0}^{4} \exp(x_k^2) \underbrace{\frac{\int_0^1 L_4^{(k)}(x)\,dx}{L_4^{(k)}(x_k)}}_{c_k} = \sum_{k=0}^{4} \exp(x_k^2) c_k.$$

Задача сводится к вычислению коэффициентов c_k, зависящих только от набора узлов x_0, x_1, x_2, x_3, x_4, но не от интегрируемой функции.

$$c_0 = \frac{\int_0^1 L_4^{(0)}(x)\,dx}{L_4^{(0)}(x_k)} = \frac{\int_0^1 (x-x_1)(x-x_2)(x-x_3)(x-x_4)\,dx}{(x_0-x_1)(x_0-x_2)(x_0-x_3)(x_0-x_4)} =$$

$$= \frac{\int_0^1 (x-\frac{1}{4})(x-\frac{1}{2})(x-\frac{3}{4})(x-1)\,dx}{(0-\frac{1}{4})(0-\frac{1}{2})(0-\frac{3}{4})(0-1)} = \frac{\int_0^1 [x^4 - \frac{5}{2}x^3 + \frac{35}{16}x^2 - \frac{25}{32}x + \frac{3}{32}]\,dx}{\frac{3}{32}} =$$

$$= \frac{7}{90} \approx 0{,}078.$$

$$c_1 = \frac{\int_0^1 L_4^{(1)}(x)\,dx}{L_4^{(1)}(x_k)} = \frac{\int_0^1 (x-0)(x-\frac{1}{2})(x-\frac{3}{4})(x-1)\,dx}{(\frac{1}{4}-0)(\frac{1}{4}-\frac{1}{2})(\frac{1}{4}-\frac{3}{4})(\frac{1}{4}-1)} =$$

$$= \frac{\int_0^1 [x^4 - \frac{9}{4}x^3 + \frac{13}{8}x^2 - \frac{3}{8}x]\,dx}{-\frac{3}{128}} = \frac{16}{45} \approx 0{,}356.$$

$$c_2 = \frac{\int_0^1 L_4^{(2)}(x)\,dx}{L_4^{(2)}(x_k)} = \frac{\int_0^1 (x-0)(x-\frac{1}{4})(x-\frac{3}{4})(x-1)\,dx}{(\frac{1}{2}-0)(\frac{1}{2}-\frac{1}{4})(\frac{1}{2}-\frac{3}{4})(\frac{1}{2}-1)} =$$

$$= \frac{\int_0^1 [x^4 - 2x^3 + \frac{19}{16}x^2 - \frac{3}{16}x]\,dx}{\frac{1}{64}} = \frac{2}{15} \approx 0{,}133.$$

Так как узлы равноотстоящие, то $c_3 = c_1$ и $c_4 = c_0$. Заметим, что все $c_k > 0$, $k = 0, 1, 2, 3, 4$. Это гарантирует устойчивость полученной квадратурной формулы.

$$\int_0^1 \exp(x^2)\,dx \approx \exp(0^2)c_0 + \exp[(1/4)^2]c_1 + \exp[(1/2)^2]c_2 + \exp[(3/4)^2]c_3 + \\ + \exp(1^2)c_4 \approx 1{,}463.$$

Погрешность равна

$$\Psi = \int_0^1 [\exp(x^2) - P_4(x)]\,dx = \int_0^1 \frac{[\exp(x^2)]^{(5)}|_{x=\xi}}{5!}\omega_5(x)\,dx, \text{ где } \xi \in [0,1].$$

Найдём 5-ю производную

$$(e^{x^2})' = e^{x^2}\cdot 2x, \qquad (e^{x^2})^{(\mathrm{IV})} = e^{x^2}\cdot(12 + 48x^2 + 16x^4),$$
$$(e^{x^2})'' = e^{x^2}\cdot(2 + 4x^2), \qquad (e^{x^2})^{(\mathrm{V})} = e^{x^2}\cdot(120x + 160x^3 + 32x^5).$$
$$(e^{x^2})''' = e^{x^2}\cdot(12x + 8x^3).$$

Очевидно $M_5 = \max\limits_{[0,1]} |(e^{x^2})^{(\mathrm{V})}| = e\cdot(120 + 160 + 32) = 312e$. Оценим погрешность сверху[1]

$$|\Psi| \leqslant \frac{M_5}{120}\int_0^1 |\omega_5(x)|\,dx = \frac{312e}{24}\cdot\frac{19}{3}\left(\frac{1}{4}\right)^6 \approx 0{,}05.$$

Модуль пришлось поставить внутри интеграла, а не снаружи, так как $\int_0^1 \omega_5(x)\,dx = 0$ (при равноотстоящих узлах с шагом $x_i - x_{i-1} = 1/4$ функция $\omega_5(x)$ нечётна относительно точки $x = 1/2$). Реальная погрешность при этом на один порядок (в 10 раз) оказалась меньше теоретической. □

5.6 Иинтеграл $I = \int\limits_0^1 \exp(x^2)\,dx$ был вычислен с помощью составной формулы трапеций с различными шагами h:

[1] Сейчас и в дальнейшем будем использовать следующую легко получаемую таблицу

$$\int_{-h}^{h} |\omega_3(x)|\,dx = \tfrac{1}{2}h^4, \qquad \int_{-2h}^{2h} |\omega_5(x)|\,dx = \tfrac{19}{3}h^6,$$
$$\int_{-3h}^{3h} |\omega_7(x)|\,dx = \tfrac{639}{4}h^8, \quad \int_{-4h}^{4h} |\omega_9(x)|\,dx = \tfrac{37186}{3}h^{10}.$$

h	$S_{\text{тр}}(h)$
1/32	1,46309
1/64	1,46276
1/128	1,46268
1/256	1,46266

Какая погрешность содержится в ответе на при шаге $h = 1/64$? При каком шаге достигается точность $\varepsilon = 10^{-5}$?

Решение. Для формулы трапеций справедливо равенство

$$I = S_{\text{тр}}(h) + O(h^2) = S_{\text{тр}}(h) + ch^2 + O(h^3), \text{ где } c = -\tfrac{f'(b)-f'(a)}{12} = \text{const}.$$

Отбросим малое слагаемое $O(h^3)$ и вычислим приближённо интеграл I с шагами h и $h/2$

$$I \approx S_{\text{тр}}(h) + ch^2, \quad I \approx S_{\text{тр}}(h/2) + c(h/2)^2.$$

Отсюда $c(h/2)^2 \approx [S_{\text{тр}}(h) - S_{\text{тр}}(h/2)]/3$. Погрешность равна

$$|I - S_{\text{тр}}(h/2)| \approx \frac{|S_{\text{тр}}(h) - S_{\text{тр}}(h/2)|}{3}.$$

По исходным данным составим таблицу

$h/2$	$\lvert S_{\text{тр}}(h) - S_{\text{тр}}(h/2)\rvert/3$
1/64	0,00011
1/128	0,00003
1/256	0,00001

Из полученных данных следует, что $h = 1/64$ погрешность будет порядка 10^{-4}. Для достижения точности $\varepsilon = 10^{-5}$ следует выбрать шаг $h = 1/256$.

□

2.6 Занятие 6

2.6.1 Численное интегрирование (продолжение)

6.1 Оценить минимальное число разбиений отрезка N для вычисления интеграла $I = \int_0^1 \sin(x^2)dx$ по составной квадратурной формуле трапеций $S(h)$, обеспечивающее точность $\varepsilon = 10^{-4}$.

Решение. По условию погрешность $|\Psi| = |I - S(h)| \leqslant \varepsilon$. Для составной формулы трапеций $S(h) = h\sum\limits_{i=1}^{n} \frac{f(x_{i-1})+f(x_i)}{2}$ справедливо $|\Psi| \leqslant \frac{M_2(b-a)}{12}h^2$. Очевидно, число разбиений отрезка N и длина частичного отрезка h связаны соотношением $h = (b-a)/N$. Оценим M_2. В производной $\frac{d^2}{dx_2}\sin(x^2) = \underbrace{-4x^4\sin(x^2)}_{f(x)} + \underbrace{2\cos(x^2)}_{g(x)}$ оба слагаемых убывают на отрезке $[0,1]$. Следовательно, максимум модуля производной будет достигаться на границе отрезка $M_2 = \max(2, |-4\sin 1 + 2\cos 1|) \approx 2{,}29$.

В итоге получаем $\frac{M_2(b-a)}{12}h^2 \leqslant \varepsilon$ или $\frac{M_2(b-a)^3}{12N^2} \leqslant \varepsilon$. Так как $N \in \mathbb{N}$, то $N \geqslant \left[\sqrt{\frac{M_2(b-a)^3}{12\varepsilon}}\right] + 1$, где $[x]$ означает целую часть числа x. Окончательно $N \geqslant \left[\sqrt{\frac{2{,}29}{12\cdot 10^{-4}}}\right] + 1 = [43{,}7] + 1 = 44$. □

6.2 Предложить способ вычисления интеграла $\int_0^1 \frac{\ln x}{1+x^2}dx$ по составной квадратурной формуле с постоянным шагом h.

Решение. Интеграл является несобственным, так как $\lim\limits_{x\to 0+0}\frac{\ln x}{1+x^2} = -\infty$. Заметим, что для любого $\alpha > 0$

$$\lim_{x\to 0+0} x^\alpha \ln x = \lim_{x\to 0+0}\frac{\ln x}{x^{-\alpha}} \overset{\text{прав. Лопит.}}{=} \lim_{x\to 0+0}\frac{x^{-1}}{-\alpha x^{-\alpha-1}} = 0.$$

$$\int_0^1 \frac{\ln x}{1+x^2}dx = \int_0^1 \frac{(1+x^2-x^2)\ln x}{1+x^2}dx = \int_0^1 \ln x\,dx - \int_0^1 \frac{x^2\ln x}{1+x^2}dx$$

Первый интеграл вычисляется явно

$$\int_0^1 \ln x\,dx = x\ln x|_0^1 - \int_0^1 x\frac{1}{x}\,dx = -1.$$

Второй интеграл уже не является несобственным, следовательно, для его вычисления применима, например, составная формула Симпсона. □

6.3 Построить квадратурную формулу для вычисления интеграла $\int_1^\infty \frac{f(x)}{1+x^2}dx$, где $|f(x)| \leqslant B = \text{const}$ с заданной точностью ε.

Решение. Интеграл является несобственным так как верхний предел равен ∞. Разобьём его на две части

$$\int_1^\infty \frac{f(x)}{1+x^2}dx = \int_A^\infty \frac{f(x)}{1+x^2}dx + \int_A^\infty \frac{f(x)}{1+x^2}dx = I_1 + I_2.$$

Подберём число A достаточно большим, чтобы $|I_2| \leqslant \varepsilon/2$. Функция $f(x)$ по условию ограничена некоторой константой B. Будем считать, что значение B нам известно

$$|I_2| = \left|\int_A^\infty \frac{f(x)}{1+x^2}dx\right| \leqslant \int_A^\infty \frac{B}{1+x^2}dx = B\,\mathrm{arctg}\,x|_A^\infty = B\left(\frac{\pi}{2} - \mathrm{arctg}\,A\right).$$

$$B\left(\frac{\pi}{2} - \mathrm{arctg}\,A\right) \leqslant \frac{\varepsilon}{2} \;\Rightarrow\; A \geqslant \mathrm{tg}\left(\frac{\pi}{2} - \frac{\varepsilon}{2B}\right).$$

□

В итоге по заданным ε и B находим значение A. Далее, например, с помощью составной формулы Симпсона вычисляем $I_1 = \int_1^A \frac{f(x)}{1+x^2}dx$ с точностью $\frac{\varepsilon}{2}$. Общая погрешность результата составит $\frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon$.

2.6.2 Матричные вычисления

6.4 Доказать, что для любых матриц A и B и фиксированной векторной нормы $||\mathbf{x}||$ для подчинённой матричной нормы справедливо $||AB|| \leqslant ||A|| \cdot ||B||$.

6.5 Показать, как связаны между собой нормы $||\mathbf{x}||_1$, $||\mathbf{x}||_2$ и $||\mathbf{x}||_\infty$.

Решение. □

6.6 Найти матричные нормы, подчинённые векторным нормам $||\mathbf{x}||_1$, $||\mathbf{x}||_2$ и $||\mathbf{x}||_\infty$.

Решение. Для любого вектора $\mathbf{x} \in \mathbb{R}^n$ справедливо

$$||A\mathbf{x}||_\infty = \max_i \left|\sum_{j=1}^n a_{ij}x_j\right| \leqslant \max_i \left(\sum_{j=1}^n |a_{ij}| \max_j |x_j|\right) \leqslant \max_i \left(\sum_{j=1}^n |a_{ij}|\right) ||\mathbf{x}||_\infty.$$

Покажем, что эта оценка достигается. Пусть максимум по i имеет место при $i = k$. Тогда для $\mathbf{x} = (\mathrm{sign}(a_{k1}), \mathrm{sign}(a_{k2}), \ldots, \mathrm{sign}(a_{kn}))^\top$ имеем $||\mathbf{x}||_\infty = 1$ и точные равенства по всей цепочке выше. Таким образом, $||A||_\infty = \max\limits_i \left(\sum_{j=1}^n |a_{ij}|\right)$.

Аналогично показывается, что $||A||_1 = \max_j \left(\sum_{i=1}^n |a_{ij}|\right)$.

По определению

$$||A||_2 = \sup_{\mathbf{x}\neq 0} \frac{||A\mathbf{x}||_2}{||\mathbf{x}||_2} = \sup_{\mathbf{x}\neq 0} \sqrt{\frac{(A\mathbf{x}, A\mathbf{x})}{(\mathbf{x},\mathbf{x})}} = \sup_{\mathbf{x}\neq 0} \sqrt{\frac{(A^\top A\mathbf{x}, \mathbf{x})}{(\mathbf{x},\mathbf{x})}}.$$

Матрица $B = A^\top A$ — симметричная и $(B\mathbf{x}, x) = (A\mathbf{x}, A\mathbf{x}) \geqslant 0$, следовательно, (во-первых) все её собственные значения $\lambda_i(B) \geqslant 0$, а (во-вторых) сама матрица B обладает ортонормированной системой собственных векторов $\mathbf{q}_1, \ldots, \mathbf{q}_n$. Это означает, что $B\mathbf{q}_i = \lambda_i \mathbf{q}_i$ и $(\mathbf{q}_i, \mathbf{q}_j) = \begin{cases} 1, & \text{если } i = j \\ 0, & \text{если } i \neq j \end{cases}$. Любой вектор $\mathbf{x}$ представим в виде $\mathbf{x} = \sum\limits_{i=1}^n c_i \mathbf{q}_i$, $||\mathbf{x}||_2^2 = \sum\limits_{n=1}^n c_i^2$, поэтому $(B\mathbf{x}, \mathbf{x}) = (\sum_{i=1}^n \lambda_i c_i \mathbf{q}_i, \sum_{i=1}^n c_i \mathbf{q}_i) = \sum_{i=1}^n \lambda_i c_i^2$. Отсюда $(B\mathbf{x}, \mathbf{x}) \leqslant \max\limits_i \lambda_i ||\mathbf{x}||_2^2$. Получаем $\sup\limits_{\mathbf{x}\neq 0} \sqrt{\frac{(B\mathbf{x},\mathbf{x})}{(\mathbf{x},\mathbf{x})}} = \max\limits_i \lambda_i(B)$, а равенство достигается на соответствующем максимальному λ_i собственном векторе. Поэтому $||A||_2 = \sqrt{\max\limits_i \lambda_i(A^\top A)}$. □

6.7 Найти матричные нормы, подчинённые векторным нормам $||\mathbf{x}||_1$, $||\mathbf{x}||_2$ и $||\mathbf{x}||_\infty$ для матрицы

$$A = \begin{pmatrix} -1 & 3 & 2 \\ 3 & -3 & 4 \\ 1 & 9 & 7 \end{pmatrix}.$$

Решение. Воспользуемся результатом предыдущей задачи

$$||A||_\infty = \max_i \left(\sum_{j=1}^n |a_{ij}|\right) =$$
$$= \max\left(|-1| + |3| + |2|, |3| + |-3| + |4|, |1| + |9| + |7|\right) = 17.$$

$$||A||_1 = \max_j \left(\sum_{i=1}^{n} |a_{ij}| \right) =$$

$$= \max\left(|-1| + |3| + |1|, |3| + |-3| + |9|, |2| + |4| + |7|\right) = 15.$$

$||A||_2 = \sqrt{\max_i \lambda_i(A^\top A)}$. Найдём матрицу

$$B = A^\top A = \begin{pmatrix} 11 & -3 & 17 \\ -3 & 99 & 57 \\ 17 & 57 & 69 \end{pmatrix}$$

И её собственные значения

$$|B - \lambda E| = \begin{vmatrix} 11-\lambda & -3 & 17 \\ -3 & 99-\lambda & 57 \\ 17 & 57 & 69-\lambda \end{vmatrix} = -\lambda^3 + 179\lambda^2 - 5132\lambda + 4356.$$

Из уравнения $|B - \lambda E| = 0$ получаем $\lambda_1 \approx 143{,}43$, $\lambda_2 \approx 34{,}69$ и $\lambda_3 \approx 0{,}88$. Выбираем максимальное с.з. λ_1 и получаем $||A||_2 = \sqrt{\lambda_1} \approx 11{,}96$.

□

6.8 Можно ли утверждать, что если определитель матрицы мал, то матрица плохо обусловлена?

Решение. Пусть $D = \varepsilon E$, где $\varepsilon > 0$ — малое число и E — единичная матрица. Определитель $\det(D) = \varepsilon^n$ весьма мал, тогда как матрица D хорошо обусловлена, поскольку $\mu(D) = ||D||\,||D^{-1}|| = 1$.

Рассмотрим теперь матрицу

$$A = \begin{pmatrix} 1 & -1 & -1 & \cdots & -1 \\ 0 & 1 & -1 & \cdots & -1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix},$$

у которой определитель равен 1, и вычислим её число обусловленности.

Для этого построим в явном виде обратную матрицу.

$$A^{-1} = \begin{pmatrix} 1 & 1 & 2 & 4 & \cdots & 2^{n-3} & 2^{n-2} \\ 0 & 1 & 1 & 2 & \cdots & 2^{n-4} & 2^{n-3} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 & 1 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 \end{pmatrix},$$

$||A^{-1}||_\infty = 1 + 1 + 2 + 2^2 + \ldots + 2^{n-2} = 2^{n-1}$, $||A||_\infty = n$ и $\mu_\infty(A) = n2^{n-1}$, т.е. матрица A плохо обусловлена, хотя $\det(A) = 1$.

□

2.7 Занятие 7

2.7.1 Решение дифференциальных уравнений

7.1 Проверить, аппроксимирует ли разностная схема уравнение

$$y'(x) = f(x, y(x))$$

а) $\dfrac{1}{3h}(y_k - y_{k-3}) = f_{k-1}$;

б) $\dfrac{1}{8h}(y_k - 3y_{k-2} + 2y_{k-3}) = \dfrac{1}{2}(f_{k-1} + f_{k-2})$;

в) $\dfrac{1}{2h}(3y_k - 4y_{k-1} + y_{k-2}) = f_k$.

7.2 Для задачи $y' + y = x + 1$, $y(0) = 0$ рассматривается схема

$$\frac{y_{k+1} - y_{k-1}}{2h} + y_k = kh + 1, \quad y_0 = 0, \quad y_1 = 0.$$

Каков порядок аппроксимации у данной схемы? Можно ли его улучшить?

7.3 Для задачи $y' + 5y = 5$, $y(0) = 2$ построена разностная схема

$$\frac{y_{k+1} - y_{k-1}}{2h} + 5y_k = 5, \quad y_0 = 2, \quad y_1 = 2 - 5h.$$

Исследовать её аппроксимацию.

7.4 Построить аппроксимацию второго порядка для заданной краевой задачи

$$\begin{cases} u''(x) + u(x) = \cos x + 1, \\ u(1) = 2, \\ u'(3) - 3u(3) = 1, \end{cases}$$

Решение. Как видно из граничных условий, решение ищется на отрезке $[1, 3]$. Разобьём этот отрезок на n равных частей. Длина каждого частичного отрезка равна $h = (3-1)/n$. Введём в рассмотрение сетку $\omega = \{x_0, x_1, \ldots, x_n\}$, где $x_0 = 1, x_1 = 1+h, x_2 = 2+2h, \ldots, x_{n-1} = 3-h, x_n = 3$. Рассмотрим также сеточную функцию $y_i = y(x_i)$, определённую только в узловых точках x_i, $i = 0, 1, \ldots, n$.

Требуется построить систему линейных уравнений с неизвестными y_i, где $y_i \approx u(x_i)$. Совокупность таких y_i будет численным решением краевой задачи.

Погрешность аппроксимации должна по условию задачи иметь порядок 2 (т.е., например, уменьшение шага h вдвое должно уменьшать погрешность решения $|y_i - u(x_i)|$ в четыре раза).

Заменим в уравнении производную разделённой разностью (см. лекцию 5) $u''(x_i) \approx \frac{y_{i-1}-2y_i+y_{i+1}}{h^2}$. Исходное дифференциальное уравнение перейдёт в разностное

$$u''(x) + u(x) = \cos x + 1 \;\to\; \frac{y_{i-1} - 2y_i + y_{i+1}}{h^2} + y_i = \cos x_i + 1.$$

Убедимся, что получен второй порядок аппроксимации. В разностное уравнение вместо сеточной функции подставим точное решение, т.е. заменим y_i на $u_i = u(x_i)$. Кроме этого разложим $u_{i\pm1}$ в ряд Тейлора в окрестности точки x_i

$$u_{i\pm1} = u_i \pm hu_i' + \frac{h^2}{2}u_i'' \pm \frac{h^3}{3!}u_i''' + \frac{h^4}{4!}u^{\mathrm{IV}}(\xi^{\pm}),\ \xi^- \in [x_{i-1}, x_i], \xi^+ \in [x_i, x_{i+1}].$$

После сокращения имеем

$$u_i'' - \frac{h^2}{12}[u^{\mathrm{IV}}(\xi^-) + u^{\mathrm{IV}}(\xi^+)] + u_i = \cos x_i + 1.$$

Вычитая из последнего равенства исходное дифференциальное уравнение, получаем невязку $\psi_n^{(1)} = -\frac{h^2}{12}[u^{\mathrm{IV}}(\xi^-) + u^{\mathrm{IV}}(\xi^+)] = -\frac{h^2}{6}u^{\mathrm{IV}}(\eta) = O(h^2)$, где $\eta \in [\xi^-, \xi^+]$. Получен требуемый порядок аппроксимации.

Очевидно, замена граничного условия $u(1) = 2$ на $y_0 = 2$ не имеет погрешности аппроксимации.

В правом граничном условии нельзя заменить $u'(3)$ левой разделённой разностью, так как $u'(3) = \frac{y_n - y_{n-1}}{h} + O(h^1)$. Выделим в $O(h^1)$ главный член. Из формулы Тейлора в окрестности $x_n = 3$ имеем

$$u(3-h) = u(3) - hu'(3) + \frac{h^2}{2}u''(3) + O(h^3),$$

откуда

$$u'(3) = \frac{u(3) - u(3-h)}{h} + \frac{h}{2}u''(3) + O(h^2).$$

Из исходного уравнения следует, что $u''(3) + u(3) = \cos(3) + 1$. Таким образом,

$$\frac{u(3) - u(3-h)}{h} + \frac{h}{2}[\cos(3) + 1 - u(3)] + O(h^2) - 3u(3) = 1.$$

Невязка для правого граничного условия равна $O(h^2)$, что даёт второй порядок аппроксимации.

Окончательный ответ — это система из $n+1$ линейного уравнения с неизвестными $y_0, y_1, \ldots, y_n$

$$\begin{cases} \dfrac{y_{i-1} - 2y_i + y_{i+1}}{h^2} + y_i = \cos x_i + 1, \text{ где } i = 1, 2, \ldots, n-1 \text{ и } x_i = 1 + ih, \\ y_0 = 2, \\ \dfrac{y_n - y_{n-1}}{h} + \frac{h}{2}[\cos(3) + 1 - y_n] - 3y_n = 1. \end{cases}$$

□

Глава 3

Лабораторный практикум

3.1 ЛР 1. Распространение ошибок в вычислительных процедурах.

При построении математической модели и получении результата, на ошибку полученного ответа влияют несколько факторов:

1. *Ошибка математической модели* — ошибка неучтенных параметров физического явления. Например, полет тела, брошенного под углом к горизонту с малыми скоростями можно описывать обычными баллистическими уравнениями без учёта сопротивления воздуха. Как только скорости становятся значительными нужно вводить сопротивление воздуха. Далее следуют: учёт плотности, температуры воздуха, ветра, переменного ускорения свободного падения, вращения Земли и т.д. Построение адекватной математической модели описывается в курсе математического моделирования.

2. *Ошибка входных данных* — это ошибка измерений параметров физической модели. Во многих физических и технических задачах данная погрешность достигает нескольких процентов. Она приводит к так называемой неустранимой погрешности, которая не управляется математически и, зачастую, приводит к парадоксальным результатам.

3. *Ошибка численного метода* — связана с тем, что исходные операторы заменяются приближёнными. Например, интеграл – суммой, дифференцирование – конечной разностью, функцию – полиномом, бесконечный ряд – конечной суммой элементов. Погрешность численного

метода обычно берут в 2-5 раза меньше неустранимой погрешности. Меньше брать невыгодно из-за увеличения объёма выполняемых вычислений, больше — из-за снижения точности вычислений. Ошибка численного метода управляется математически.

4. *Ошибка округления* связана с ограниченной разрядной сеткой компьютера, из-за чего, например, бесконечные десятичные дроби представляются в конечном виде.

③ Начнём со второго пункта. Рассмотрим многочлен Уилкинсона

$$P(x) = (x-1)(x-2)...(x-20) = x^{20} - 210x^{19} + 20615x^{18} + \dots$$

Очевидно, корнями его являются $x_1 = 1$, $x_2 = 2, \dots$, $x_{20} = 20$. Выполните в MATLAB команду `p=poly(1:20)`, которая позволяет получить коэффициенты полинома, корнями которого является аргумент функции. Наберите `roots(p)` и убедитесь, что корни полинома найдены верно. Измените значение, например, второго коэффициента на малую величину 10^{-7} (составляет $\approx 5 \cdot 10^{-8}\%$ от числа) и снова выполните эту команду - половина корней стала комплексными числами! Исходная задача оказалась неустойчивой к входным данным (их малое изменение ведет к сильному изменению решения), в результате чего получился абсурдный результат.

③ Рассмотрим представление числа в компьютере. Чаще всего в MATLAB производятся операции над числами с двойной точностью. Число формата double — 64-разрядное число (8 байт), в котором 1 бит — знаковый, 52 отводятся под мантиссу и 11 под порядок числа. ($D = \pm(1+m)\cdot 2^n$, $m = 0, m_1 m_2 \dots m_k$, $m_1 \neq 0$ — мантисса числа, n — порядок). Так как в порядке тоже есть знаковый бит, то множество его значений лежит от $-2^{10} + 1 = -1023$ до $2^{10} = 1024$. В командном окне MATLAB наберите `2^1023`. Получилось число с десятичным порядком +308 (его легко оценить аналитически: $2^{1024} \approx 2^{1000} = 1024^{100} \approx (10^3)^{100} = 10^{300}$). Теперь выполните `2^1024` — получили `Inf` т.е. машинную бесконечность. Команда `realmax` как раз и выдаёт максимальное число, которое можно представить в MATLAB, команда `realmin` — минимальное (по абсолютной

величине).

③ Теперь посмотрим на мантиссу: $2^{52} \approx 4,5 \cdot 10^{15}$, таким образом, мантисса содержит 15-16 десятичных знаков; все, что вылезет за эти пределы, будет отброшено. Установите формат отображения с плавающей точкой: `format long e`, выполните `sqrt(2)` (квадратный корень из 2). Посчитайте число выданных цифр.

Точность	Байты	M_0 (маш.ноль)	M_∞ (маш. бесконечность)
Одинарная	4	$1,2 \cdot 10^{-38}$	$3,4 \cdot 10^{38}$
Двойная	8	$2,2 \cdot 10^{-308}$	$1,8 \cdot 10^{308}$

③ Казалось бы, такая точность представления способна удовлетворить любые нужды исследователя. Однако, необходимо помнить, что, например, перед тем как два числа будут сложены они должны быть приведены к единому порядку, а то, что при сложении выйдет за мантиссу будет отброшено! Поэтому, если сложить `10^8+10^-7` в мантиссу уложатся 15 десятичных знаков и будет получен верный результат, а если выполнить `10^8+10^-8`, то малое число выйдет за разрядную сетку, и получим те же `10^8` (проверьте).

③ Получается парадоксальная ситуация: давайте прибавим к 1 малое число `10^-16`, но последовательно `10^17` раз[1](такие вычисления характерны для задач ЧМ — вычисление рядов, разнообразные итерационные, разностные задачи...). Легко найти ответ 11, однако, машинная арифметика даёт 1 — ошибка 1000%! Если бы мы начали с конца, т.е. сначала нашли сумму малых, а затем прибавили к единице, то получили бы верный результат. Таким образом, машинное сложение, в общем случае, не коммутативно.

③ Знание этих фактов позволяет протестировать компьютер на погрешность вычислении. Найдем значение выражения $\frac{1+\varepsilon-1}{\varepsilon}$ при $\varepsilon = 2^{-n}$, $n = 0, 1, 2, \ldots$. С точки зрения аналитической математики значение этого

[1]Цикл из 10^{17} итераций на компьютерах невысокой производительности может вычисляться долго. Чтобы убедиться в округлении мантиссы достаточно меньшего количества итераций, например 10^6. Прервать длительные вычисления в MATLAB можно клавишами `Ctrl+Break`.

выражения постоянно и равно единице для любых n. Сделайте предположения, при каком значении n это выражение перестанет отличаться от единицы, чему будет равно; напишите программу, реализующую этот алгоритм с пошаговой выдачей номера шага, значения ε и результата выражения.

④ Рассмотрим погрешности численных методов. Построим алгоритм вычисления интеграла $I_n = \int_0^1 x^n e^{x-1}\,dx$, $n = 1, 2, 3, \ldots$ Интегрируя по частям, находим:

$$I_1 = \int_0^1 xe^{x-1}\,dx = xe^{x-1}|_0^1 - \int_0^1 e^{x-1}\,dx = \tfrac{1}{e}$$
$$I_2 = \int_0^1 x^2e^{x-1}\,dx = x^2e^{x-1}|_0^1 - 2\int_0^1 xe^{x-1}\,dx = 1 - 2I_1$$
$$\ldots$$
$$I_n = \int_0^1 x^ne^{x-1}\,dx = x^ne^{x-1}|_0^1 - n\int_0^1 x^{n-1}e^{x-1}\,dx = 1 - nI_{n-1}$$

Вычислите значения интегралов, до $n = 30$. Заметьте, что подинтегральная функция на всем отрезке интегрирования неотрицательна, следовательно, и значение интеграла — положительное число. Более того, подынтегральная функция на данном интервале ограничена функцией $y = 1$, т.е. значение интеграла не может превышать единицы. Посмотрите, как это согласуется с полученными результатами. Попытайтесь на основе полученных сведений о погрешностях объяснить данный феномен.

⑤ Напишите функцию вычисления значения синуса в виде конечной суммы ряда[2] ($\sin x = \sum_{k=0}^{\infty}(-1)^k \frac{x^{2k+1}}{(2k+1)!}$). По признаку Лейбница погрешность вычисления сходящегося знакопеременного ряда не превышает по абсолютной величине первого из отброшенных членов. Вычисления членов ряда проводите до вычисления члена по модулю не превышающего

[2]При работе со степенными рядами на компьютере для сокращения объёма вычислений полезно рассмотреть выражение u_{n+1}/u_n. Например, в случае синуса: $u_n = (-1)^n \frac{x^{2n+1}}{(2n+1)!}$ и $u_{n+1} = (-1)^{n+1}\frac{x^{2n+3}}{(2n+3)!}$, откуда $\frac{u_{n+1}}{u_n} = \frac{-x^2}{(2n+2)(2n+3)}$. Получаем, $u_0 = x$, $u_1 = u_0 \cdot \frac{-x^2}{(2\cdot 0+2)(2\cdot 0+3)}$, $u_2 = u_1 \cdot \ldots$ и т.д. Такой подход позволяет обойтись без вычисления «лишних» степеней и факториалов, а следовательно, избежать больших чисел в числителе и знаменателе.

10^{-17}. Вычислите значение синуса в точках $0, \frac{\pi}{3}, \frac{\pi}{2}, \pi, 2\pi$, убедитесь, что полученные значения с высокой степенью точности совпадают с действительным значением синуса в этих точках. Далее проведите вычисления в точках $12\pi, 13\pi, 14\pi$, выводя результаты по шагам (посчитанный член и частичную сумму ряда). Полученные результаты объясняются погрешностями округления, в реальных программах значение аргумента приводится к отрезку $[0; \frac{\pi}{2}]$.

3.2 ЛР 2. Методы дихотомии, Ньютона, простых итераций.

Решение данных уравнений подразумевает два этапа:

1. Локализация корней — выделение отрезков, на которых находится не более одного корня.

2. Поиск корня с заданной точностью.

В качестве функции $f(x)$ рассмотрим полином: $f(x) = x^3 - 3x^2 - 9x - 5$. Известно, что корни полинома $P_n(x) = \sum_{k=0}^{n} a_k x^k$ (в общем случае комплексные) лежат внутри круга $|x_p| \leqslant 1 + \frac{1}{|a_n|} \max(|a_0|, |a_1|, \ldots, |a_{n-1}|)$. Мы только что применили аналитический подход к сужению области нахождения корней.

③ Напишите файл-функцию `f.m` и постройте график функции на отрезке $[-10; 10]$, включив командой `grid on` отображение линий сетки. Выделите отрезки, содержащие нули функции (графический способ это один из методов локализации корней). Очевидно, функция имеет корни одинарной и двойной кратности. Запишите вектор `p`, содержащий коэффициенты полинома, и найдите его корни, выполнив команду `roots(p)`.

③ Напишите программу, реализующую нахождение корня одинарной кратности методом деления отрезка пополам. Обратите внимание, что метод дихотомии предполагает, что значения функции на концах отрезка различаются по знаку. Выведите на экран число итераций.

③ Напишите программу нахождения решений уравнения $f(x) = 0$ методом Ньютона и используйте её для поиска всех корней полинома. Вы-

ведите на экран число итераций.

④ Для кратного корня использовать модифицированный метод Ньютона. Выведите на экран число итераций.

③ Найдем методом простых итераций корни уравнения $x^2 - a = 0$ (квадратный корень из числа a). Приведем уравнение к виду, удобному для использования метода: $x = \frac{1}{2}\left(\frac{a}{x} + x\right)$. Можно убедиться, что правая часть уравнения удовлетворяет условию сходимости метода (в отличие от таких представлений как: $x = x^2 + x - a$, $x = \frac{a}{x}$). Напишите программу вычисления квадратного корня с машинной точностью.

④ Исследовать область сходимости представления $x = x^2 + x - a$. Произвести расчёт в найденной области и за её пределами.

③ В MATLAB для решения уравнений вида $f(x) = 0$ есть функция `fzero`, в качестве параметров которой передаётся имя файл-функции и начальное приближение корня (или отрезок его содержащий). Обратите внимание, что `fzero` так же как и метод дихотомии требует, чтобы при переходе через корень функция меняла знак (например, с её помощью не удастся найти нули функции, $f(x) = \sin x + 1$, корни полинома двойной кратности и т.д.)

⑤ Сделайте предположения о том, где находятся корни уравнения $\sin x = x/2$ и найдите их, используя все изученные методы.

Реализация функциями MATLAB

`fzero` — поиск нулей функции

`roots` — поиск корней полинома

Контрольные вопросы

1. Из каких соображений, и какими методами можно локализовать искомый корень?

2. Каким образом реализуется заданная точность поиска в методе половинного деления и в методе Ньютона? Чем различаются условия

прекращения итераций?

3. Почему с помощью метода половинного деления не удаётся находить корни двойной кратности?

4. Сравните (перечислите преимущества и недостатки) методов Ньютона и половинного деления.

5. Назовите условия применимости метода Ньютона.

6. Оцените скорость сходимости в методе Ньютона при поиске корней одинарной и двойной кратности.

7. Назовите условия сходимости метода простых итераций.

8. Каким образом можно априорно вычислить примерное количество итераций, требуемых для нахождения корня с заданной точностью, для всех изученных методов?

3.3 ЛР 3. Интерполяция функций. Полиномы Лагранжа, Ньютона.

Пусть есть прибор, который в дискретные моменты времени выдаёт сигнал по закону $f(t) = \sin \pi t$. Допустим, наблюдатель зарегистрировал пять отсчётов в моменты времени $t_i = \frac{i}{4}$, $i = 0, 1, 2, 3, 4$. Задачей наблюдателя (который не знает закона выдачи сигнала) является получение приближённого значения функции на отрезке $[0, 1]$ в любой момент времени.

③ Используя линейную интерполяцию, найдите значения функции в точках: $t = 0, \frac{1}{6}, \frac{1}{3}, \frac{1}{2}$ и сравните с реальным значением синуса в этих точках. Постройте графики синуса и ломаной, проходящей через пять заданных точек. Отметьте, насколько сильно они различаются в разных частях графика. Чем это обусловлено?

③ Постройте по заданным пяти точкам интерполяционный многочлен Лагранжа или Ньютона и, используя его, найдите значения функции в точках $t = 0, \frac{1}{6}, \frac{1}{3}, \frac{1}{2}$, Сравните результаты со значениями, полученными

при линейной интерполяции, и значениями синуса в этих точках. Постройте графики синуса и интерполяционного многочлена. Какую максимальную ошибку мы допускаем при аппроксимации синуса данным полиномом? Сравните экспериментальную погрешность с теоретической.

④ В программе сделать возможность строить многочлен Лагранжа или Ньютона для произвольного набора точек $t = t_0, t_1, \ldots, t_n$.

⑤ При вычислении многочлена стараться заменить циклы матричными операциями (см. первое практическое занятие).

③ Найдите значение интерполяционного полинома при $t = 2$. Почему оно так сильно отличается от значения синуса в этой точке?

④ Задайте функцию Рунге $f(x) = \frac{1}{1+25x^2}$ на отрезке $[-5, 5]$ в десяти равноотстоящих точках. Сравните значения функции и интерполяционного полинома при $x = 4,5$. Постройте графики функции и полинома на заданном отрезке и объясните поведение интерполяционного полинома. Посмотрите, что будет происходить при постепенном увеличении числа узлов интерполяции и подумайте, как можно избавиться от получившегося эффекта.

⑥ Для приближения функции Рунге используйте Чебышёвские узлы. Постройте графики функции и многочлена.

Реализация функциями MATLAB. Для одномерной интерполяции в MATLAB есть функция `interp1`. Изучите её и постройте графики интерполированных функций из примеров, рассмотренных выше.

Контрольные вопросы

1. Системами каких функций можно приближать заданную таблично функцию? Из каких соображений выбирается эта система? Приведите примеры.
2. Чем различается построение интерполяционных полиномов Лагранжа и Ньютона?
3. Сколько полиномов и какой степени можно провести через n точек?

4. Пусть таблично заданно достаточное количество точек некоторой степенной функции. Возможно ли и как восстановить коэффициенты этого многочлена?

5. Каким образом за счёт выбора узлов можно добиться уменьшения ошибки интерполяции?

6. Выпишите формулы для оценки погрешности интерполяции в точке и на отрезке.

7. Что называется кусочной интерполяцией и каковы критерии её применимости?

8. Каким образом следует поступить, если ставится не прямая, а обратная задача: требуется найти значение x, при котором $f(x)$ принимает заданное значение?

3.4 ЛР 4. Дифференцирование функции, заданной таблично.

③ Выберите некоторую функцию (например, $\sin x$, $\cos x$, $\exp x$, $\operatorname{sh} x$, $\operatorname{ch} x$, $\ln x$, ...) и некоторую точку x из области определения функции. Найдите значение производной функции в выбранной точке (используя любую формулу численного дифференцирования) с точностью 10^{-3}, 10^{-6}. Пользоваться точным значением производной в качестве эталона запрещено[3].

④ Выберите некоторую функцию (например, $\sin x$, $\cos x$, $\exp x$, $\operatorname{sh} x$, $\operatorname{ch} x$, $\ln x$, ...) и некоторую точку x из области определения функции. Сравните погрешности у формул с разными порядками погрешностей (например, $y'(x) \approx \frac{y(x+h)-y(x)}{h}$ и $y'(x) \approx \frac{y(x+h)-y(x-h)}{2h}$) для последовательности убывающих шагов (например, $h = \frac{1}{2}, \frac{1}{4}, \frac{1}{8}$). С какими скоростями убывают погрешности для каждой формулы? Дайте теоретическую оценку и подтвердите ответ экспериментом[4].

[3] В данном задании лучше использовать формулу $y'(x) \approx \frac{f(x)-f(x-h)}{h}$ или формулу $y'(x) \approx \frac{f(x+h)-f(x-h)}{2h}$. Погрешность для этих формул легко получить из разложения $f(x \pm h)$ в ряд Тейлора в окрестности точки x.

[4] Напомним, что $y'(x) = \frac{y(x+h)-y(x)}{h} + O(h^1)$ и формула имеет первый порядок погрешности; $y'(x) = \frac{y(x+h)-y(x-h)}{2h} + O(h^2)$ и формула имеет второй порядок погрешности.

⑤ Неустойчивость численного дифференцирования. Выберите некоторую функцию (например, $\sin x$, $\cos x$, $\exp x$, $\operatorname{sh} x$, $\operatorname{ch} x$, $\ln x$, $\dots$) и некоторую точку x из области определения функции. Попробуйте применить формулу $y'(x) \approx \frac{y(x+h)-y(x)}{h}$ для стремящейся к нулю последовательности $h = \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \dots$). Будет ли погрешность $\varepsilon = \left| y'(x) - \frac{y(x+h)-y(x)}{h} \right|$ монотонно убывать при уменьшении h? Сравните практический и теоретический результаты.

Реализация функциями MATLAB.

`p = polyfit(x,y,n)` — вычисление коэффициентов полинома наилучшего (среднеквадратического) приближения степени n.

`k = polyder(p)` — получение коэффициентов k полинома, получающегося при дифференцировании полинома, заданного коэффициентами p.

`y = polyval(p,x)` — вычисление значения полинома с коэффициентами `p` в точках `x`.

Контрольные вопросы

1. Как теоретически узнать погрешность формулы численного дифференцирования? Как узнать порядок погрешности?
2. Какие есть способы получения формул численного дифференцирования?
3. Какие есть способы практической (при вычислении на компьютере) оценки погрешности численного дифференцирования?
4. Являются ли формулы численного дифференцирования устойчивыми к погрешностям входных данных? Ответ обоснуйте.
5. Опишите, как имея в распоряжении формулу для численного дифференцирования с порядком точности p, получить формулу с большим порядком точности (метод Рунге).

3.5 ЛР 5. Интегрирование функций. Формулы трапеций, Симпсона.

③ Задайте функцию $f(x) = x^3$ на отрезке $[0, 1]$. Очевидно, определённый интеграл от функции $f(x)$ на этом отрезке равен $\frac{1}{4}$. Напишите программу, вычисляющую значение интеграла по формулам трапеций и Симпсона. Какую максимальную теоретическую ошибку мы при этом допускаем? Найдите реальное значение погрешности (абсолютное значение разности между теоретическим и аналитическим решением). Почему при вычислении интеграла по формуле Симпсона от данной функции ошибка равна нулю? Какие бы получились значения погрешностей для квадратичной и линейной функций (предположите и проведите численный эксперимент для $f_2(x) = x^2$, $f_1(x) = x/2$ на отрезке $[0, 1]$).

④ Используя соотношение $\int_0^1 \frac{1}{1+x^2} dx = \text{arctg}(1)$ найдите значение числа π с точностью 10^{-6}. В данном задании в процессе вычислений нельзя использовать встроенную константу `pi` для определения величины шага. Из каких соображений выбирался шаг для получения указанной точности?

⑤ Реализовать предыдущее задание, определяя точность методом Рунге. При численном вычислении интегралов последовательно с шагами h и $h/2$ можно сократить число арифметических операций. Заметим, что приближённое значение интеграла $I_{h/2}$ есть сумма, часть слагаемых которой возможно уже участвовало при вычислении I_h. Поэтому можно получить $I_{h/2}$, используя числовое значение I_h. Это позволяет избежать повторного суммирования части слагаемых[5].

[5] Продемонстрируем это на примере метода трапеций для шагов h и $h/2$:

$$I_h = h\left(\frac{1}{2}f_0 + f_1 + \ldots + f_{n-1} + \frac{1}{2}f_n\right),$$

$$I_{h/2} = \frac{h}{2}\left(\frac{1}{2}f_0 + f_{1/2} + f_1 + f_{3/2} + \ldots + f_{n-1} + f_{n-1/2} + \frac{1}{2}f_n\right).$$

Очевидно, что $I_{h/2} = \frac{I_h}{2} + \frac{h}{2}\left(f_{1/2} + f_{3/2} + \ldots + f_{n-3/2} + f_{n-1/2}\right)$.

Реализация функциями MATLAB. `q = quad(fun,a,b)` — вычисление интеграла от функции `fun` (встроенной или описанной файл-функцией) на отрезке $[a, b]$. Алгоритм основан на квадратурной формуле Симпсона с автоматическим подбором шага.

Контрольные вопросы

1. В каких случаях имеет смысл использовать неравномерное распределение узлов? Каким образом алгоритмически можно реализовать автоматический подбор шага?
2. Какая ошибка допускается, если подынтегральная функция заменяется интерполяционным полиномом, а затем производится аналитическое вычисление интеграла?
3. Какой метод — прямоугольников (с выбором центральной точки) или трапеций — даёт в общем случае меньшую ошибку?
4. Каким образом можно уточнить значение интеграла, уже вычисленного по формулам трапеций и прямоугольников?

3.6 ЛР 6. Решение систем линейных уравнений.

③ Задайте матрицу A и вектор-столбец f системы линейных уравнений $AX = f$, используя генератор случайных чисел. Очевидно, можно получить решение таким образом: $X = A^{-1}f$ (предварительно проверив, что матрица A не вырожденная) или по правилу Крамера ($x_i = \frac{\det A_i}{\det A}$, где A_i — матрица, получающаяся из матрицы A заменой i-го столбца на столбец правой части f). Реализуйте и проверьте работоспособность этих методов. Несмотря на простоту использования в MATLAB, эти варианты чрезвычайно неэкономичны по числу операций.

③ Напишите программу нахождения решения системы линейных уравнений методом Гаусса с выбором главного элемента.

③ Функция `rref` MATLAB также приводит матрицу `[A f]` к диагональному виду, из которого сразу же видно решение системы. Также па-

кет содержит операцию левого матричного деления, с помощью которой очень просто найти решение: `X = A\f`. Более того, эта операция позволяет решать недоопределённые и переопределённые системы линейных уравнений, выбирая алгоритм решения в зависимости от вида матрицы A.

③ Задайте случайным образом матрицу A размерности 20×20 и вектор X. Определите число обусловленности матрицы A с помощью функции `cond`. Изменяя значения некоторых элементов матрицы A, добейтесь, чтобы её число обусловленности стало больше 10^3. Используя A и X, найдите вектор $f = AX$. Полагая вектор X неизвестным, решите систему линейных уравнений всеми предложенными выше методами и сравните найденные решения с уже известным. Какой из методов дал более точный результат? Обратите внимание на решения, полученные обычным методом Гаусса и методом с выбором главного элемента.

Реализация функциями Matlab.

`\` — операция левого матричного деления

`rref(A)` — приведение матрицы к диагональному виду

`inv(A)` — нахождение обратной матрицы

`cond(A)` — нахождение числа обусловленности матрицы

3.7 ЛР 7. Метод Эйлера. Схемы Рунге-Кутта решения ОДУ.

Рассмотрим обыкновенное дифференциальное уравнение p-го порядка:

$$y^{(p)} = f(x, y, y', y'', \ldots, y^{(p-1)})$$

Путём введения замены, данное уравнение можно свести к системе линейных уравнений первого порядка:

$$\begin{cases} y_{k+1} = y'_k, \quad k = 1, 2, \ldots, p-1, \\ y'_p = f(x, y_1, y_2, \ldots, y_p), \end{cases}$$

где $y_1 = y$. Данная система может быть записана в векторной форме:

$$\frac{d\overline{y}}{dx} = \overline{f}(x, \overline{y})$$

Для получения единственного решения из системы нужно наложить p дополнительных условий на функции $y_k(x)$. Для задачи Коши данные условия задаются в одной точке: $y_k(x_0) = \eta_k$, $k = 1, 2, \ldots, p$. Эти условия рассматриваются как задание начальной точки для интегральной кривой в $(p+1)$-мерном пространстве $(x, y_1, y_2, \ldots, y_p)$. Если правые части системы непрерывны и ограничены в некоторой окрестности начальной точки $(x_0, \eta_1, \eta_2, \ldots, \eta_p)$, то решение задачи Коши существует, но может быть не единственно. Если правые части к тому же удовлетворяют условию Липшица по переменным y_k, то решение существует и единственно, т.е. задача Коши поставлена корректно.

Рассмотрим уравнение 1-го порядка:

$$\begin{cases} \frac{dy}{dx} = f(x, y) \\ y(x_0) = \eta \end{cases}$$

и пусть данная задача Коши поставлена корректно. Будем искать численное решение уравнения на отрезке $[x_0, X]$. Введем на этом отрезке сетку $\overline{\omega}_h = \{x_i, i = 0, 1, \ldots, N\}$, таким образом, чтобы $x_0 < x_1 < \ldots < x_N = X$. Обозначим $h_i = x_{i+1} - x_i$, $i = 0, 1, \ldots, N-1$ шаг сетки. Заменив производную в уравнении правой разностью, получим

$$\frac{y_{i+1} - y_i}{h_i} = f_i, \qquad i = 0, 1, \ldots, N-1,$$

где $f_i = f(x_i, y_i)$.

Зная $y(x_0) = \eta$, можно найти все остальные значения y_i по формуле: $y_{i+1} = y_i + h_i f_i$, $i = 0, 1, \ldots, N-1$. Данный метод нахождения численного решения называется методом Эйлера (или методом ломаных). Схемы, в которых значение функции явно выражается через уже найденные значения, называются явными, иначе - неявными. Таким образом, схема Эйлера является явной. Оценка погрешности для данного метода дает $O(\max(h_i))$, что предполагает малый шаг сетки для получения удовлетворительного решения.

③ Найдите численное решение следующего ОДУ методом Эйлера (на равномерной сетке) и сравните его с аналитическим:

$$\begin{cases} \frac{dy}{dx} = x^2, \\ y(0) = 1. \end{cases}$$

④ MATLAB имеет множество функций для численного решения обыкновенных дифференциальных уравнений и их систем. Солверы `ode23` и `ode45` основаны на формулах Рунге-Кутты 2,3 и 4,5 порядков соответственно. Разберем пример их использования на примере задачи о колебаниях под воздействием внешней силы:

$$\begin{cases} y'' + 2y' + 10y = \sin t, \\ y(0) = 1,\ y'(0) = 0. \end{cases}$$

Сводим к системе уравнений первого порядка:

$$\begin{cases} y_1' = y_2, \\ y_2' = -2y_2 - 10y_1 + \sin t, \\ y_1(0) = 1,\ y_2(0) = 0. \end{cases}$$

```
function test ode:
Y0 = [1;0]; % вектор начальных условий
[T Y] = ode45('oscil',[0 15],Y0); % получение решения
                                  % на отрезке 0<t<15
function F=oscil(t,y)             % составляем функцию
F=[y(2); -2*y(2)-10*y(1)+sin(t)]; % из правых частей
```

вектор `Y(:,1)` содержит решение исходного уравнения,
вектор `Y(:,2)` содержит производную решения уравнения.

④ Постройте графики координаты $y_1(t)$ и скорости $y_2(t)$. Воспользовавшись знаниями теории обыкновенных дифференциальный уравнений можно получить аналитическое решение:

$$y = e^{-t}(C_1 \cos 3t + C_2 \sin 3t) + \frac{1}{85}(9 \sin t - 2 \cos t),$$

где для данной задачи Коши $C_1 = \frac{87}{85}$, $C_2 = \frac{26}{85}$. Постройте график аналитического решения и сравните с численным, полученным при помощи `ode23` и `ode45`.

⑤ Решите следующее дифференциальное уравнение:

$$\begin{cases} y'' = -\frac{1}{t^2}, \\ y(t_0) = \ln t_0, \quad \text{при } t_0 = 0,01 \end{cases}$$

и сверьте численное решение с аналитическим $y = \ln t$.

Реализация функциями MATLAB. `ode45`, `ode23`, `ode113`, `ode15s`, `ode23s`, `ode23t`, `ode23tb`.

Глава 4

Приложения

4.1 Список тем для реферативно-расчётной работы

1. Нахождение всех корней (в том числе комплексных) произвольного многочлена степени $\leqslant 20$ методом *парабол*.
   Литература: [4] гл.5, §2, п.8.
2. Интерполяция сплайнами (вычисления методом прогонки)
   Литература: [4] гл.2, §1, п.9; [5] гл.3, §4.
3. Интерполяция многочленами Эрмита
   Литература: [4] гл.2, §1, п.6; [5] гл.3, §3.
4. Интегрирование методом Симпсона с автоматическим выбором шага
   Литература: [5] гл.4, §1, п.5.
5. Решение краевой задачи для дифференциального уравнение 2-го порядка с граничным условием 1-го рода методом прогонки
6. Решение краевой задачи для дифференциального уравнение 2-го порядка с граничным условием 2-го рода методом прогонки
7. Решение краевой задачи для дифференциального уравнение 2-го порядка с граничным условием 3-го рода методом прогонки
8. Решение системы линейных уравнений методом Якоби
9. Решение системы линейных уравнений методом Зейделя
10. Решение системы линейных уравнений методом вращений

11. Решение системы линейных уравнений методом LU-разложений
12. Вычисление обратной матрицы методом LU-разложений
13. Численное решение дифференциальных уравнений методом Рунге-Кутты 4-го порядка
14. Интегрирование методом Гаусса Литература: [5] гл.4, §3.
15. Решение системы нелинейных уравнений методом Ньютона Литература: [5] гл.5, §4.
16. Численное решение дифференциальных уравнений методом Адамса Литература: [5] гл.6, §3.
17. Поиск собственных значений матрицы степенным методом
18. Решение системы линейных уравнений методом релаксации
19. Решение системы линейных уравнений методом наискорейшего градиентного спуска

4.2 Определитель Вандермонда

Будем вычислять *определитель Вандермонда* индуктивно, избавляясь на k-ом шаге от x_k. Вначале имеем:

$$\Delta(x_1, x_2, \ldots, x_n) = \begin{vmatrix} 1 & 1 & \cdots & 1 \\ x_1 & x_2 & \cdots & x_n \\ x_1^2 & x_2^2 & \cdots & x_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{n-1} & x_2^{n-1} & \cdots & x_n^{n-1} \end{vmatrix}.$$

Вычитая первый столбец из всех последующих и разложив полученный определитель по первой строке имеем:

$$\Delta(x_1,x_2,\ldots,x_n)=\begin{vmatrix} 1 & 0 & \cdots & 0 \\ x_1 & x_2-x_1 & \cdots & x_n-x_1 \\ x_1^2 & x_2^2-x_1^2 & \cdots & x_n^2-x_1^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{n-1} & x_2^{n-1}-x_1^{n-1} & \cdots & x_n^{n-1}-x_1^{n-1} \end{vmatrix}=$$

$$=\begin{vmatrix} x_2-x_1 & x_3-x_1 & \cdots & x_n-x_1 \\ x_2^2-x_1^2 & x_3^2-x_1^2 & \cdots & x_n^2-x_1^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_2^{n-1}-x_1^{n-1} & x_3^{n-1}-x_1^{n-1} & \cdots & x_n^{n-1}-x_1^{n-1} \end{vmatrix}.$$

Теперь вычтем из каждой строки предыдущую, умноженную на x_1:

$$\Delta(x_1,x_2,\ldots,x_n)=$$

$$=\begin{vmatrix} x_2-x_1 & x_3-x_1 & \cdots & x_n-x_1 \\ x_2(x_2-x_1) & x_3(x_3-x_1) & \cdots & x_n(x_n-x_1) \\ \vdots & \vdots & \ddots & \vdots \\ x_2^{n-2}(x_2-x_1) & x_3^{n-2}(x_3-x_1) & \cdots & x_n^{n-2}(x_n-x_1) \end{vmatrix}.$$

Вынесем из каждого столбца общий множитель:

$$\Delta(x_1,x_2,\ldots,x_n)=(x_2-x_1)(x_3-x_1)\ldots(x_n-x_1)\begin{vmatrix} 1 & 1 & \cdots & 1 \\ x_2 & x_3 & \cdots & x_n \\ \vdots & \vdots & \ddots & \vdots \\ x_2^{n-2} & x_3^{n-2} & \cdots & x_n^{n-2} \end{vmatrix}=$$

$$=(x_2-x_1)(x_3-x_1)\ldots(x_n-x_1)\Delta(x_2,\ldots,x_n).$$

С определителем $\Delta(x_2,\ldots,x_n)$ выполним такие же вычисления:

$$\Delta(x_1,x_2,\ldots,x_n)=$$
$$=(x_2-x_1)(x_3-x_1)\ldots(x_n-x_1)\cdot(x_3-x_2)\ldots(x_n-x_2)\Delta(x_3,\ldots,x_n)$$

Продолжив вычисления получим:

$$\Delta(x_1, x_2, \ldots, x_n) = (x_2 - x_1)(x_3 - x_1)\ldots(x_n - x_1)\cdot \\ \cdot (x_3 - x_2)\ldots(x_n - x_2)\ldots(x_n - x_{n-1}) = \prod_{i<j}(x_j - x_i).$$

4.3 Ряд Тейлора

Теорема 11.2. *Если функция $f(x)$ имеет $n+1$ производную на отрезке с концами a и x, то для произвольного положительного числа p справедливо, что*

$$f(x) = \sum_{k=0}^{n} \frac{f^{(k)}(a)}{k!}(x-a)^k + R_{n+1}(x),$$

где остаточный член $R_{n+1}(x)$ может быть представлен в одной из следующих форм:

1. *форма Шлёмильха—Роша:*
   $R_{n+1}(x) = \left(\frac{x-a}{x-\xi}\right)^p \frac{(x-\xi)^{n+1}}{n!p} f^{(n+1)}(\xi)$, *где* $\xi \in [a, x]$;
2. *форма Лагранжа:*
   $R_{n+1}(x) = \frac{(x-a)^{n+1}}{(n+1)!} f^{(n+1)}(\xi)$, $\xi \in [a, x]$;
3. *форма Коши:*
   $R_{n+1}(x) = \frac{(x-a)^{n+1}(1-\theta)^n}{n!} f^{(n+1)}[a + \theta(x-a)]$, *где* $0 < \theta < 1$;
4. *интегральная форма:*
   $R_{n+1}(x) = \frac{1}{n!}\int\limits_a^x (x-t)^n f^{(n+1)}(t)\, dt$;
5. *форма Пеано:*
   $R_{n+1}(x) = o[(x-a)^n]$.

4.4 Модифицированный метод Ньютона

Покажем, что модифицированный метод Ньютона имеет квадратичную скорость сходимости.

Пусть уравнение $f(x) = 0$ имеет корень x_* кратности p. Последнее по определению кратного корня означает, что

$$f(x_*) = f'(x_*) = f''(x_*) = \ldots = f^{(p-1)}(x_*) = 0, \text{ но } f^{(p)}(x_*) \neq 0.$$

Из обоих частей модифицированного метода Ньютона

$$x_{n+1} = x_n - \frac{pf(x_n)}{f'(x_n)}$$

отнимем x_*

$$x_{n+1} - x_* = x_n - x_* - \frac{pf(x_n)}{f'(x_n)} = \frac{(x_n - x_*)f'(x_n) - pf(x_n)}{f'(x_n)}.$$

Если обозначить $F(x) = (x - x*)f'(x) - pf(x)$, то числитель можно заменить на $F(x_n)$:

$$x_{n+1} - x_* = \frac{F(x_n)}{f'(x_n)}. \tag{4.1}$$

Несложно заметить, что

$$\begin{aligned}
&F(x) = (x - x_*)f'(x) - pf(x) && \Rightarrow \quad F(x_*) = 0,\\
&F'(x) = (x - x_*)f''(x) - (p-1)f'(x) && \Rightarrow \quad F'(x_*) = 0,\\
&F''(x) = (x - x_*)f'''(x) - (p-2)f''(x) && \Rightarrow \quad F''(x_*) = 0,\\
&\ldots && \quad \ldots\\
&F^{(p-1)}(x) = (x - x_*)f^{(p)}(x) -\\
&\qquad -(p - (p-1))f^{(p-1)}(x) && \Rightarrow \quad F^{(p-1)}(x_*) = 0,\\
&F^{(p)}(x) = (x - x_*)f^{(p+1)}(x) -\\
&\qquad -(p-p))f^{(p)}(x) = (x - x_*)f^{(p+1)}(x) && \Rightarrow \quad F^{(p)}(x_*) = 0.
\end{aligned}$$

Разложим теперь $F(x_n)$ в ряд Тейлора с центром в точке x_*. Остаточный член запишем в интегральной форме.

$$\begin{aligned}
F(x_n) = \underbrace{F(x_*)}_{=0} + \underbrace{F'(x_*)}_{=0}(x_n - x_*) + \underbrace{F''(x_*)}_{=0}\frac{(x_n - x_*)^2}{2!} + \ldots +\\
+ \underbrace{F^{(p-1)}(x_*)}_{=0}\frac{(x_n - x_*)^{p-1}}{(p-1)!} + \frac{1}{(p-1)!}\int\limits_{x_*}^{x_n} F^{(p)}(t)(x_n - t)^{p-1}\,dt =\\
= \frac{1}{(p-1)!}\int\limits_{x_*}^{x_n} F^{(p)}(t)(x_n - t)^{p-1}\,dt.
\end{aligned}$$

Вспомним, что $F^{(p)}(t) = (t - x_*)f^{(p+1)}(t)$. Тогда

$$F(x_n) = \frac{1}{(p-1)!}\int\limits_{x_*}^{x_n}(x_n - t)^{p-1}(t - x_*)f^{(p+1)}(t)\,dt.$$

Так как функция $(x_n - t)^{p-1}(t - x_*)$ не меняет знак на отрезке $[x_*, x_n]$, то можно применить теорему о среднем и вынести $f^{(p+1)}(t)$ из-под знака интеграла:

$$F(x_n) = \frac{f^{(p+1)}(\xi_n)}{(p-1)!}\int\limits_{x_*}^{x_n}(x_n - t)^{p-1}(t - x_*)\,dt, \quad \text{где } \xi_n \in [x_*, x_n].$$

Легко получить, что

$$\int\limits_{x_*}^{x_n}(x_n - t)^{p-1}(t - x_*)\,dt = \frac{(x_n - x_*)^{p+1}}{p(p+1)}.$$

В итоге

$$F(x_n) = \frac{f^{(p+1)}(\xi_n)}{(p-1)!}\cdot\frac{(x_n - x_*)^{p+1}}{p(p+1)} = \frac{f^{(p-1)}(\xi_n)(x_n - x_*)^{p+1}}{(p+1)!}.$$

Числитель выражения (4.1) получен, рассмотрим теперь знаменатель.

Разложим $f'(x_n)$ в ряд Тейлора с центром разложения x_*. Остаточный член ряда запишем в форме Лагранжа:

$$\begin{aligned} f'(x_n) = \underbrace{f'(x_*)}_{=0} + \underbrace{f''(x_*)}_{=0}(x_n - x_*) + \underbrace{f'''(x_*)}_{=0}\frac{(x_n - x_*)^2}{2!} + \ldots + \\ + \underbrace{f^{(p-1)}(x_*)}_{=0}\frac{(x_n - x_*)^{(p-2)}}{(p-2)!} + f^{(p)}(\eta_n)\frac{(x_n - x_*)^{(p-1)}}{(p-1)!} = \\ = f^{(p)}(\eta_n)\frac{(x_n - x_*)^{(p-1)}}{(p-1)!}, \quad \text{где } \eta_n \in [x_*, x_n]. \end{aligned}$$

Вернёмся теперь к (4.1):

$$\begin{aligned} x_{n+1} - x_* = \frac{F(x_n)}{f'(x_n)} = \left(\frac{f^{(p+1)}(\xi_n)(x_n - x_*)^{p+1}}{(p+1)!}\right)\Bigg/\left(f^{(p)}(\eta_n)\frac{(x_n - x_*)^{(p-1)}}{(p-1)!}\right) = \\ = \frac{1}{p(p-1)}\cdot\frac{f^{(p-1)}(\xi_n)}{f^{(p)}(\eta_n)}(x_n - x_*)^2. \end{aligned}$$

Обозначим погрешность приближения $|x_{n+1} - x_*|$ на шаге $n+1$ через ε_{n+1} и аналогично $\varepsilon_n = |x_n - x_*|$. Пусть

$$M_{p+1} = \max_x |f^{(p+1)}(x)|, \qquad m_p = \min_x |f^{(p)}(x)|.$$

В итоге справедлива оценка:

$$\varepsilon_{n+1} \leqslant \frac{M_{p+1}}{p(p-1)m_p} \cdot \varepsilon_n^2.$$

Литература

[1] *Косарев В. И.* 12 лекций по вычислительной математике (вводный курс) // М.: Издательство МФТИ, 2000.

[2] *Турчак Л. И., Плотников П. В.* Основы численных методов. Учебное пособие. // М.: Физматлит, 2005

[3] *Бахвалов Н. С., Лапин А. В., Чижонков Е. В.* Численные методы в задачах и упражнениях // М.: Бином, 2010.

[4] *Калиткин Н. Н.* Численные методы // М.: Наука, 1978.

[5] *Самарский А. А., Гулин А. В.* Численные методы // М.: Наука, 1989.