Численные методы

Василий Алфёров по лекциям Евгения Яревского

2 февраля 2019 г.

Содержание

1. Погрешности

Различают абсолютную и относительную погрешности.

Абсолютная погрешность – это модуль разности значения и экспериментальных данных. Обозначают её как ΔA или, иногда, ∇A.

Относительная погрешность – это $\frac{\Delta A}{A}$. Чем ближе к нулю, тем интереснее смотреть на относительную погрешность, а не на абсолютную.

Интересный пример большой относительной погрешности: после 1995 года масса нейтрино считалась как $-22 \pm 17_{stat} \pm 17_{syst}$ эВ^2. Сейчас уже измерили точнее (Нобелевская премия по физике 2015 года).

В английском языке погрешность обозначается словом “error”, имеющим нейтральную окраску. В русском языке слово “ошибка” имеет негативную окраску, поэтому чаще используют слово “погрешность”.

Источники погрешностей (ошибок):

A) Ошибки входных данных. Делятся на:

- Случайные. Подразумевают, что никакую величину в реальном мире нельзя измерить абсолютно точно.
- Систематические. Это либо приборные ошибки (подразумевают, что идеальных приборов не существует), либо погрешности в самом методе измерения.

B) Ошибки представления действительных чисел в компьютере и округления арифметических операций.

C) Ошибки из-за “обрезания” бесконечно малых и бесконечно больших величин.

D) Упрощения в математических моделях.

E) Человеческие и машинные ошибки. К машинным ошибкам можно отнести, например, историю с Pentium.

Ошибки типов A и D никак не контролируются, с ними приходиться смириться. Ошибки типа C, как правило, могут контролироваться. Ошибки типа B могут контролироваться частично.

Ошибки точно измерить нельзя, иначе бы они были не ошибками, а поправками. Поэтому обычно их оценивают сверху.

1.1. Представление действительных чисел в компьютере

Стандарт IEEE-754. Подразумевается, что мы всё это уже знаем, поэтому остановимся только на основных моментах.

Во-первых, мы можем сохранить в наших типах лишь конечное количество чисел. Из этого следует, например, что корректные с компьютерной точки зрения числа не образуют никакой алгебраической структуры. Если мы можем представить числа a и b, то мы не обязательно можем представить $a + b$. То же верно и для любой другой арифметической операции. Если $a + b = a$, то не обязательно $b = 0$. И у нас нет ни ассоциативности, ни дистрибутивности. Коммутативность, однако, есть.

Пример.
Иллюстрация отсутствия ассоциативности при одинарной точности (`float` в плюсах):

$$\sum_{n=1}^{10^9} \frac{1}{n} = 15.4036827087 \qquad \sum_{n=10^9}^{1} \frac{1}{n} = 18.8079185486$$

~~Не знаю, на каком пентиуме считались числа в презентации, у меня получилось вот так.~~ Настоящее значение равно при этом 21.3004815023. В реальной жизни используются, в основном, числа двойной точности (`double` в плюсах и джаве, `float` в питоне). Видимо, по поводу того, что `long double` из коробки есть только в плюсах, в остальных языках за ним надо лезть в неочевидные библиотеки.

***Определение* 1.1.**
Математически эквивалентные алгоритмы – алгоритмы, эквивалентные в предположении, что используется точная арифметика.

***Определение* 1.2.**
Вычислительно эквивалентные алгоритмы – алгоритмы, эквивалентные при использовании машинной арифметики с небольшой погрешностью.

Это не одно и то же.

Пример.
Вычислим e^{-10}, используя одинарную точность, двумя способами:

$$e^{-10} = \sum_{k=0}^{N} \frac{(-10)^k}{k!} = -6.25618267804 \cdot 10^{-5} \qquad e^{-10} = \left(\sum_{k=0}^{N} \frac{10^k}{k!} \right)^{-1} = 4.5399923671 \cdot 10^{-5}$$

Как видно, вычислительно способы не эквивалентны, хотя математически оба ряда сходятся к e^{-10}. Настоящее значение равно $4.53999297624849 \cdot 10^{-5}$. Очень большая погрешность в первом способе объясняется тем, что в начале мы попеременно складываем положительные и отрицательные слагаемые, далёкие по модулю от нуля.

Иногда, чтобы представлять порядок ошибки, используют интервальную арифметику.

1.2. Статистические оценки погрешности

Максимальные оценки погрешности зачастую пессимистичны, ведь они не учитывают знак. Обычно всё же ошибки друг друга компенсируют. Альтернативой является статистический анализ.

В рамках статистического анализа обычно считается, что ошибки независимы и случайны, хотя это выполняется и не всегда.

Пример.
Пусть каждое значение x_i имеет погрешность $|\Delta x_i| \leqslant \delta$. Тогда максимальная погрешность их суммы $y = \sum x_i$ оценивается как

$$|\Delta y| \leqslant \sum_{i=1}^{n} |\Delta x_i| \leqslant n\delta$$

Пусть теперь числа при операциях округляются (не усекаются, то есть нету перекоса от округления вниз и матожидание ошибок равно нулю). Пусть также мы считаем, что дисперсия

каждой из ошибок x_i ограничена сверху числом ε. Тогда дисперсия ошибки их суммы будет оценена как

$$D[\Delta y] \leqslant \sqrt{\sum_{i=1}^{n} \varepsilon^2} = \varepsilon\sqrt{n}$$

Эмпирически хорошо работает правило: если максимальная ошибка оценивается как $uf(n)$, то дисперсия будет оцениваться как $u\sqrt{f(n)}$. Для того, чтобы это работало, обязательно требуется, чтобы матожидание ошибки было нулевым.

1.3. Переполнение и потеря точности

Переполнение – превышение максимальных допустимых значений (на минуточку, у `double` это порядка $1.8 \cdot 10^{308}$). Возникает, например, при попытке вычислить модуль вектора или комплексного числа~~, когда оба компонента комплексного числа или вектора имеют порядок 10^{154}, видимо~~. Предлагаемый способ борьбы: заранее вынести большую константу из-под корня.

Потеря точности – существенное уменьшение числа значащих цифр в процессе вычислений. Например, возникает при вычитании близких больших чисел. Способы борьбы: считать производные или приводить аргументы функций к малым диапазонам.

1.4. Распространение ошибок

Входные данные, как мы уже выяснили, в вычислительных задачах, как правило, неточные. В ходе вычислений их погрешности эволюционируют и приводят к погрешности результата. В этом разделе мы обсудим конкретные оценки ошибок.

Теорема 1.1 (Сложение и вычитание)**.**
Пусть у величин x_1, ..., x_n известны максимальные погрешности $|\Delta x_1|$, ..., $|\Delta x_n|$.

Тогда у величины $y = \sum\limits_{i=1}^{n} x_i$ максимальная погрешность оценивается как $|\Delta y| \leqslant \sum\limits_{i=1}^{n} |\Delta x_i|$.

Доказательство.
Побуду занудой и скажу, что это неравенство треугольника для модуля. □

Теорема 1.2 (Произвольная функция одного аргумента)**.**
Пусть у величины x известная максимальная погрешность $|\Delta x|$.

Пусть также $f \in C^1[x, x + \Delta x]$.

Обозначим величину $y = f(x)$.

Тогда существует такое $\xi \in [x, x + \Delta x]$, что $|\Delta y| \leqslant |f'(\xi)\Delta x|$.

Доказательство.
По теореме Лагранжа о среднем значении, существует ξ такое, что $f'(\xi)\Delta x = f(x+\Delta x) - f(x)$. Остаётся лишь заметить, что $\Delta y = |f(x + \Delta x) - f(x)|$. □

На практике часто считают Δx достаточно маленьким и берут $\xi = x$. Однако это не работает в случае, если у f в точке x экстремум – тогда нужно писать более строгие оценки.

Теорема 1.3 (Умножение и деление)**.**
Пусть у величин x_1, ..., x_n известны максимальные относительные погрешности $|\frac{\Delta x_1}{x_1}|$, ..., $|\frac{\Delta x_n}{x_n}|$.

Тогда у величины $y = \prod\limits_{i=1}^{n} x_i^{m_i}$ максимальная относительная погрешность оценивается как $|\frac{\Delta y}{y}| \leqslant \sum\limits_{i=1}^{n} |m_i||\frac{\Delta x_i}{x_i}|$.

Доказательство.

$$\left|\frac{\Delta y}{y}\right| = |\ln' y \cdot \Delta y| \leqslant |\Delta \ln y| = \left|\Delta\left(\sum_{i=1}^{n} m_i \ln x_i\right)\right| \leqslant \sum_{i=1}^{n} m_i |\Delta \ln x_i| \leqslant \sum_{i=1}^{n} m_i \left|\frac{\Delta x_i}{x_i}\right|$$

□

Теорема 1.4 (Функция нескольких переменных)**.**
Пусть у величин x_1, ..., x_n известны максимальные погрешности Δx_1, ..., Δx_n.

Пусть также $f : \mathbb{R}^n \to \mathbb{R}^n$ – функция, непрерывно дифференцируемая на отрезке $[x; x + \Delta x]$.

Обозначим величину $y = f(x_1, \ldots, x_n)$.

Тогда существует такое $\theta \in [0, 1]$, что $|\Delta y| \leqslant \sum\limits_{i=1}^{n} \left|\frac{\partial f}{\partial x_i}(x + \theta\Delta x)\right| |\Delta x_i|$.

Доказательство.
Применим теорему 1.2 к функции $F(t) := f(x + t\Delta x)$. □

Опять же, нередко берут значения частных производных в точке x, что точно так же может оказаться неверным в экстремумах.

Статистическая погрешность в последнем случае оценивается как

$$\varepsilon \approx \sqrt{\sum_{i=1}^{n} \left(\frac{\partial f}{\partial x_i}\right)^2 \varepsilon_i^2}$$

1.5. Плохо обусловленные задачи

Если маленькие изменения во входных данных вызывают большие изменения в выходных данных, то задачу называют плохо обусловленной, иначе – хорошо обусловленной. Чем хуже обусловлена задача, тем большие требования по погрешности предъявляются к её решениям.

***Определение* 1.3.** Рассмотрим вычислительную задачу $f(x) : \mathbb{R}^n \to \mathbb{R}^m$.

Зафиксируем входные данные $\hat{x} \neq 0$ и решение $\hat{y} = f(\hat{x}) \neq 0$.

Относительным числом обусловленности мы в таком случае назовём число

$$\kappa(f, \hat{x}) = \lim_{\varepsilon \to 0} \sup_{\|h\| = \varepsilon} \left\{ \frac{\|f(x + h) - f(x)\|}{\|f(x)\|} \cdot \frac{\|x\|}{\|h\|} \right\}$$

Формально, из этого определения получится, что для достаточно малых возмущений (норма которых ограничена ε) будет выполняться

$$\|\hat{y} - y\| \leqslant \kappa\varepsilon\|y\| + (\varepsilon^2)$$

2. Ряды. Суммирование и ускорение сходимости

Напоминание.
Ряд – это сумма $\sum\limits_{n=0}^{\infty} a_n$.

Для него вводят частичные суммы $S_n = \sum\limits_{k=0}^{n} a_k$.

Под суммой ряда подразумевают предел $\lim\limits_{n\to\infty} S_n$.

Любой предел $\lim\limits_{n\to\infty} a_n$ представим в виде суммы $a_0 + \sum\limits_{n=1}^{\infty}(a_{n+1} - a_n)$.

2.1. Оценки хвоста ряда

Понятное дело, что численно сумму ряда мы обычно вычисляем как $\sum\limits_{n=0}^{N} a_n$. Вопрос в том, как выбрать N для достижения заданной точности.

Определим остаток ряда как $R_n = \sum\limits_{k=n}^{\infty} a_k$ и будем оценивать $|R_N|$. Кроме того, нужно вычислить S_N таким образом, чтобы при этом не возникло излишней погрешности, например, из-за машинной арифметики.

Теорема 2.1 (Сравнение с геометрической прогрессией)**.**
Пусть существует такое $0 < \kappa < 1$, что $\forall j \geqslant N \;\; |a_{j+1}| \leqslant \kappa |a_j|$. Тогда

$$|R_N| \leqslant \frac{|a_{N+1}|}{1-\kappa} \leqslant \frac{\kappa}{1-\kappa}|a_N|$$

Доказательство.

$$|R_N| \leqslant \sum_{j=N+1}^{\infty} \kappa^{j-1-N}|a_{N+1}| = \frac{|a_{N+1}|}{1-\kappa} \leqslant \frac{\kappa}{1-\kappa}|a_N|$$

□

Теорема 2.2 (Сравнение с интегралом)**.**

1. Пусть $\forall j \geqslant N \;\; |a_j| \leqslant f(j)$, где $f(x)$ невозрастает на $x \geqslant N$. Тогда

$$|R_N| \leqslant \sum_{j=N+1}^{\infty} |a_j| \leqslant \int_N^{\infty} f(x)dx$$

2. Пусть $\forall j \geqslant N \;\; |a_j| > g(j) > 0$. Тогда

$$|R_N| = \sum_{j=N+1}^{\infty} a_j > \int_N^{\infty} g(x)dx$$

Доказательство.
Строгого доказательства не было, да оно здесь и не нужно. Было доказательство методом пристального вглядывания в картинку.

TODO: Скопипастить картинку из презентации. □

Пример.

$$S = \sum_{i=1}^{\infty} \frac{1}{i^2}$$

Оценим функцией $f(x) = \frac{1}{x^2}$.

$$|R_n| \leqslant \int_N^{\infty} \frac{1}{x^2} dx = \frac{1}{N} \leqslant \varepsilon \Rightarrow N \geqslant \frac{1}{\varepsilon}$$

То есть для достижения точности ε достаточно сложить $\frac{1}{\varepsilon}$ членов.

Замечание.
Можно оценить снизу функцией $g(x) = \frac{1}{(x-1)^2}$ и получить оценку снизу на требуемое количество слагаемых того же порядка.

Замечание.
Так как мы работаем с достаточно большими числами, члены ряда можно (неформально) заменять эквивалентными. Например, для оценки требуемого количества членов для ряда $\frac{1}{i^3+2i^2+1}$ можно оценить лишь требуемое количество членов $\frac{1}{i^3}$, так как сильного различия между результатами не будет.

Теорема 2.3 (Знакопеременный ряд)**.**
Пусть a_n – знакопеременный ряд, такой, что $|a_n|$ монотонно убывает, S – его сумма, S_n – частичные суммы, R_n – остатки.

Тогда R_n и R_{n+1} имеют разные знаки и $S_n \leqslant S \leqslant S_{n+1}$. Более того,

$$S = \frac{1}{2}(S_n + S_{n+1}) \pm \frac{1}{2}|a_{n+1}|$$

Также выполняется $|R_n| \leqslant |a_n|$.

Доказательство.
Рассмотрим отдельно ряд чётных частичных сумм S_0, S_2, ...и нечётных S_1, S_3, ... Несложно заметить, что один из них возрастает и сходится к S, а другой убывает и сходится к S. Таким образом доказано первое предложение из утверждения теоремы.

Второе предложение утверждения теоремы: если S лежит на отрезке $[S_n; S_{n+1}]$, то S отличается от его середины не более чем на половину его длины.

Третье предложение утверждения теоремы (если $R_n \geqslant 0$, иначе симметрично): $R_n = S - S_n \leqslant S_{n+1} - S_n = a_n$. □

Замечание.
Несмотря на то, что можно придумать пример, где погрешность будет порядка $\frac{1}{2}|a_{n+1}|$, фактически она обычно много меньше.

2.2. Методы ускорения сходимости

Обычно суммирование происходит в несколько этапов:

- Выбор адекватного представления в машинной арфметике (~~с `BigDecimal` не ошибётесь~~).
- Выбор аналитического и алгоритмического представления, минимизирующего ошибки.
- Ускорение сходимости. (~~Казалось бы, это нужно сделать в предыдущий этап?~~)

Ускорение сходимости – это преобразование $\{s_n\}_{n=0}^{\infty} \to \{s'_k\}_{k=0}^{\infty}$, такое, что s'_k сходятся туда же и быстрее. Как правило, s'_k зависит от первых k элементов s_n. Такое преобразование может применяться итеративно.

Для рядов ускорение сходимости – это ускорение сходимости частичных сумм.

Модельные ряды (последовательности)
Пусть $a_n \sim b_n$, то есть $\lim\limits_{j\to\infty} \frac{a_j}{b_j} = 1$. Обозначим $c_n = a_n - b_n = a_n(1 - \frac{b_j}{a_j})$. Тогда

$$\sum_{j=1}^{\infty} a_j = S + \sum_{j=1}^{\infty}(a_j - b_j) = S + \sum_{j=1}^{\infty} c_j$$

Здесь $S = \sum\limits_{j=1}^{\infty} b_j$. В то же время можно заметить, что $c_n = a_n(1 - \frac{b_n}{a_n})$ убывают быстрее, чем a_n, откуда их ряд сходится быстрее.

Пример.

$$\sum_{j=1}^{\infty} \frac{1}{j^2} = \frac{\pi^2}{6}$$

Возьмём модельный ряд $\sum\limits_{j=1}^{\infty} \frac{1}{j(j+1)}$.

Сумма этого ряда равна $\sum\limits_{j=1}^{\infty} \frac{1}{j(j+1)} = \sum\limits_{j=1}^{\infty} \frac{1}{j} - \frac{1}{j+1} = 1$.

С помощью этого ряда можно получить ускорение (можно оценить по теореме 2.2, что требуется меньше членов для достижения той же точности):

$$\frac{\pi^2}{6} = \sum_{j=1}^{\infty} \frac{1}{j^2} = 1 + \sum_{j=1}^{\infty} \frac{1}{j^2(j+1)}$$

.

Ускорение Эйткина (Aitken)
***Определение* 2.1.**
Введём рекурсивно символ $\nabla^k b_n$:

1. $\nabla^1 b_n = \nabla b_n = b_n - b_{n-1}$
2. $\nabla^k b_n = \nabla^{k-1} b_n - \nabla^{k-1} b_{n-1}$.

В предыдущем методе мы ускоряли изначально ряды. В этом методе будем ускорять изначально пределы. Понятно, что они друг к другу сводятся.

Этот метод хорош своей универсальностью – не надо ничего придумывать. Однако в общем случае не гарантируется, что он сработает.

Основная идея: выберем моделью геометрическую прогрессию $y_n = s + b\kappa^n$. Понятное дело, она сходится к s. Мы предполагаем, что исходная последовательность s_n в каком-то смысле "похожа" на эту геометрическую прогресиию. В этом предположении мы для члена s_j исходя из двух предыдущих членов s_j и s_{j-1} подберём параметры для y_n. То есть предполагаем, что для $n = j, j-1, j-2$ выполгяется $y_n = s_n$ и подберём исходя из этого s.

Заметим: $\nabla y_n = b\kappa^{n-1}(\kappa - 1)$. Тогда

$$\frac{\nabla s_j}{\nabla s_{j-1}} = \frac{\nabla y_j}{\nabla y_{j-1}} = \kappa$$

Получаем:

$$b\kappa^j = b\kappa^{j-1} \cdot \frac{\kappa(\kappa-1)}{\kappa - 1} = \frac{b\kappa^{j-1}(\kappa-1)}{\frac{\kappa-1}{\kappa}} = \frac{\nabla s_j}{1 - \frac{1}{\kappa}} = \frac{\nabla s_j}{1 - \frac{\nabla s_{j-1}}{\nabla s_j}} = \frac{(\nabla s_j)^2}{\nabla^2 s_j}$$

С другой стороны, $b\kappa^j = s_j - s$, откуда получаем $s = s_j - \frac{(\nabla s_j)^2}{\nabla^2 s_j}$. Такую оценку мы и объявляем требуемым преобразованием:

$$s'_j = s_j - \frac{(\nabla s_j)^2}{\nabla^2 s_j}$$

Замечание.
Если s_j – геометрическая последовательность, то $s'_j \equiv s$. Это верно просто по построению последовательности.

Теорема 2.4 (Без доказательства)**.**
Пусть s_j – такая последовательность, что

1. $\lim\limits_{j\to\infty} s_j = s$
2. $\lim\limits_{j\to\infty} \frac{s_{j+1}-s_j}{s_j - s_{j-1}} = \kappa^*, \quad |\kappa^*| < 1$

Тогда последовательность $\{s'_j\}$ сходится быстрее, чем $\{s_j\}$.

Замечание.
Теорему можно переформулировать в терминах рядов, это будет выглядеть логичнее.

Теорема наиболее эффективна при $\kappa^* = -1$. **TODO:** В формулировке $|\kappa^*| < 1$, это подозрительно.

Пример.

$$S = \sum_{j=1}^{\infty} \frac{1}{j2^j}$$

Понятное дело, ряд сходится. Посмотрим на отношение соседних элементов этого ряда:

$$\frac{\frac{1}{(j+1)2^{j+1}}}{\frac{1}{j2^j}} = \frac{j}{2(j+1)} \to \frac{1}{2}$$

Отсюда по теореме 2.4 получаем, что после применения ускорения Эйткина ряд начнёт сходиться быстрее.

Построенный метод может применяться итеративно: можно построить последовательности $\{s''\}$, $\{s'''\}$, и так далее.

Кроме того, может оказаться полезной идея применить ускорение не к последовательности $\{s_j\}$, а к прореженной последовательности (s_4, s_8, s_{12} и так далее).

Экстраполяция по Ричардсону

Пусть $F(h)$ – задача с параметром h. Мы умеем вычислять $F(h)$ при $h > 0$, нас интересует значение $F(0)$. Таким параметром может быть, например, небольшой шаг при численном решении дифференциального уравнения. Пусть также выполняется

$$F(h) = a_0 + a_1 h^p + o(h^p)$$

Пусть $q > 1$. Вычислим F в точках h и qh:

$$\begin{aligned} F(h) &= a_0 + a_1 h^p + o(h^p) \\ F(qh) &= a_0 + a_1 (qh)^p + o(h^p) \\ F(0) = a_0 &= F(h) + \frac{F(h) - F(qh)}{q^p - 1} + o(h^p) \end{aligned}$$

Величина $\frac{F(h)-F(qh)}{q^p-1}$ называется поправкой Ричардсона и может использоваться как для коррекции результата, так и для оценки погрешности.

Экстраполяцию по Ричардсону также можно использовать итеративно.

Другие методы ускорения сходимости

- Преобразование Эйлера
- Усреднение частичных сумм
- Формула суммирования Эйлера-Маклорена
- Интерполяция Паде

2.3. Аппроксимация и интерполяция

***Определение* 2.2.**

Функция f аппроксимирует функцию g, если она её приближает. Обычно под этим прдразумевают, что она к ней близка по какой-то норме.

Если f совпадает с g в каком-то конечном числе точек, то f – интерполяция.

Если требуются значения f вне отрезка с заданным набором точек, то f – экстраполяция.

f – интерполирующая функция, если $f(x_i) = y_i$ для таблицы $\{x_i, y_i\}_{i=0}^N$, в которой $x_i < x_{i+1}$.

Задача интерполяции – построение интерполирующей функции, лежащей в заданном классе функций. Мы будем рассматривать только линейную интерполяцию, то есть интерполяцию в линейной оболочке заданного множества функций.

***Определение* 2.3.**
Пусть $\{\varphi_k(x)\}_{k=0}^N$ – набор линейно независимых функций на $[a;b]$. Будем считать, что все φ_k непрерывны.

Рассмотрим H – их линейную оболочку. Пусть $f \in H$ и удовлетворяет $f(x_i) = y_i$. Тогда

$$f(x_i) = y_i = \sum_{k=0}^{N} a_k \varphi_k(x_i)$$

Пусть матрица Φ образована значениями $\{\varphi_k(x_i)\}$. Тогда верхнее равенство можно записать в виде

$$\Phi^T a = f$$

Система $\{\varphi_i(x)\}$ называется чебышевской, если $\det \Phi \neq 0$.

Для чебышевской системы функций (далее ЧСФ) задача интерполяции всегда однозначно разрешима.

2.4. Интерполяция полиномами

Полиномы удобны по следующим причинам:

1. Значения полиномов в точке легко вычислять.
2. Полиномами легко оперировать.
3. Согласно теореме Вейерштрасса, они плотны в $C[a;b]$, то есть сколь угодно хорошо приближают любую непрерывную функцию.
4. Согласно теореме 2.5, они образуют ЧСФ.

Теорема 2.5.
$\{x^k\}_{k=0}^{\infty}$ – чебышевская система функций.

Доказательство.

$$\det \Phi = \prod_{1 \leqslant i < j \leqslant N} (x_j - x_i) \neq 0$$

Так как все x_i различны. Вообще, Φ – это матрица Вандермонда и её определитель хорошо известен и равен как раз написанному выше выражению. □

Проблема в том, что задача интерполяции получается плохо обусловленной, если принимать именно систему $\{x^k\}$ за базис. В качестве решения этой проблемы нередко в том же множестве выбирают альтернативный базис.

Пусть $\{p_k\}$ – базис в пространстве многочленов. Покажем, что тогда $\{p_k\}$ образуют ЧСФ. Считаем $p_j(x) = \sum c_{j,k} x^k$.

$$f(x) = \sum_{i=0}^{N} b_i p_i(x) = \sum_{j=0}^{N} \left(\sum_{k=0}^{N} b_j c_{j,k} \right) x^j = \sum_{k=0}^{N} a_k x^k$$

Здесь $C^T b = a$. Если $\det C \neq 0$, то по a всегда можно найти b. Но это и есть условие того, что $\{p_k\}$ образуют базис. Таким образом, если можно найти решение в естественном базисе, то можно найти и в базисе $\{p_k\}$.

~~А вообще, это и так понятно. И непонятно, зачем на это целый слайд городить.~~

3. Интерполяция

3.1. Интерполяционный полином в форме Лагранжа

Когда мы рассматривали Чебышевские системы функций, мы брали матрицу Ф и требовали, чтобы она была обратимой. Здесь же мы в качестве матрицы Ф возьмём вообще единичную матрицу:

$$\mathcal{L}_k(x_i) = \delta_{ki}$$

Здесь $\{\mathcal{L}_k\}$ – базис, δ_{ki} – символ Кронекера (1, если $k = i$ и 0 иначе).

Пусть $p_N(x)$ – интерполирующий полином (то есть $p_N(x_i) = y_i$). Разложим его по базису $p_N(x) = \sum\limits_{k=0}^{N} a_k \mathcal{L}_k(x)$. Заметим:

$$y_i = p_N(x_i) = \sum_{k=0}^{N} a_k \mathcal{L}_k(x_i) = a_i$$

То есть

$$p_N(x) = \sum_{k=0}^{N} y_k \mathcal{L}_k(x)$$

Научимся находить $\mathcal{L}_k$. Заметим, что у него N корней в точках x_i для $i \neq k$. Значит, он имеет вид

$$\mathcal{L}_k(x) = C_k \prod_{i \neq k} (x - x_i)$$

Ну и так как выполняется $\mathcal{L}_k(x_k) = 1$, то можно найти и C_k. Итого,

$$\mathcal{L}_k(x) = \prod_{i \neq k} \frac{x - x_i}{x_k - x_i}$$

3.2. Обусловленность полиномиальной интерполяции

Считаем, что узлы интерполяции (то есть x_i) фиксированные, а y_i известны с погрешностью $|\Delta y| \leq \varepsilon$. Берём теперь произвольную точку $\tilde{x}$ и хотим оценить, насколько изменится $p_N(\tilde{x})$ от того, что y_i мы подвигаем в пределах погрешности:

$$|\delta p(\tilde{x})| \leqslant \sum_{k=0}^{N} |\Delta y_k| \cdot |\mathcal{L}_k(\tilde{x})| \leqslant \varepsilon \sum_{k=0}^{N} |\mathcal{L}_k(\tilde{x})|$$

***Определение* 3.1** (константа Лебега)**.**

$$\Lambda_N = \max_{a \leqslant x \leqslant b} \sum_{k=0}^{N} |\mathcal{L}_k(\tilde{x})|$$

Замечание.

$$|\Delta p(\tilde{x})| \leqslant \varepsilon \Lambda_N$$

***Определение* 3.2** (Напоминание)**.**
Чебышевские многочлены определены на $[-1, 1]$, имеют вид

$$T_N(x) = \cos(N \arccos x)$$

И имеют корни в точках $x_j = \cos((2j-1)\pi/2n)$.

Теорема 3.1 (Без доказательства)**.**

1. Если x_i распределены равномерно, то

$$\Lambda_N \sim \frac{2^N}{N \log N}$$

2. Пусть мы интерполируем на $[-1; 1]$. Если $x_i = \cos((2i-1)\pi/2n)$ – корни Чебышевского многочлена T_N, то

$$\Lambda_N \leqslant 1 + \frac{2}{\pi} \log N$$

Замечание.

1. В первом случае порядок роста экспоненциальный, это очень много.
2. Во втором случае порядок роста оптимальный (наименьший) среди всех полиномиальных базисов.

3.3. Интерполяционный полином в форме Ньютона

Вообще говоря, интерполяционный полином степени N по $N+1$ точкам находится единственным образом. Вопрос в форме записи. Сейчас изучим другую.

Выберем такой базис:

$$\mathcal{N}_0(x) = 1$$
$$\mathcal{N}_k(x) = \prod_{i=0}^{k-1} (x - x_i)$$

Раз все полиномы разных степеней, то они линейно независимы.

Ищем интепроляционный полином $p(x) = \sum\limits_{k=0}^{N} a_k \mathcal{N}_k(x)$.

Есть два способа: очевидный и тот, которым пользуются. Очевидный способ: записать на эти коэффициенты систему уравнений, начиная с a_0. При этом можно воспользоваться тем, что $\mathcal{L}_k(x_j) = 0$ при $k > j$. Несложно заметить, что получается при этом треугольная система:

$$\begin{cases} a_0 = y_0 \\ a_0 + a_1(x_1 - x_0) = y_1 \\ \dots \\ \sum\limits_{k=0}^{N} a_k \mathcal{N}_k(X_N) = y_N \end{cases}$$

Для второго же введём следующие объекты, напоминающие производные:

***Определение* 3.3** (Разделённая разность)**.**
Вводим рекурсивно.

Р. р. 0-ого порядка $[x]f = f(x)$

Р. р. 1-ого порядка $[x_1, x]f = \frac{f(x)-f(x_1)}{x-x_1}$

Р. р. 2-ого порядка $[x_1, x_2, x]f = \frac{[x_1,x]f-[x_1,x_2]f}{x-x_2}$

Р. р. k-ого порядка

$$[x_1, \dots, x_{k-1}, x_k, x]f = \frac{[x_1, \dots, x_{k-1}, x]f - [x_1, \dots, x_{k-1}, x_k]f}{x - x_k}$$

Теорема 3.2.
Интерполяционный многочлен записывается в виде

$$p(x) = \sum_{k=0}^{N} [x_0, \dots, x_{k-1}, x_k] y \mathcal{N}_k(x)$$

Доказательство.
Рассмотрим разделённую разность p по узлам интерполяции и переменным x_i.

Заметим:

$$[x_0, x]p = \frac{p(x) - p(x_0)}{x - x_0} \Leftrightarrow p(x) = p(x_0) + (x - x_0)[x_0, x]p$$

Этот факт применим итеративно:

$$p(x) = p(x_0) + (x - x_0)[x_0, x]p = p(x_0) + (x - x_0)([x_0, x_1]p + (x - x_1)[x_0, x_1, x]p) =$$
$$= \dots = \sum_{k=0}^{N} [x_0, \dots, x_{k-1}, x_k]p \prod_{i=0}^{k-1} (x - x_i)$$

Осталось лишь заметить, что раз $p(x_j) = y_j \quad \forall j$, то $[x_0, \dots, x_k]p = [x_0, \dots, x_k]y$. □

TODO: Картинка с лекции с графиками базисов Лагранжа и Ньютона.

3.4. Погрешность интерполяции

Пусть мы восстанавливаем какую-то функцию f внутри отрезка $[a;b]$ по N точкам x_j. Нас интересует, насколько мы могли ошибиться.

Теорема 3.3.
Пусть $f \in C^{N+1}[a,b]$ и p_N – интерполяционный полином на узлах $\{x_i\}_{i=0}^N$. Тогда для любой $x \in [a,b]$ существует $\xi \in [a,b]$, что

$$f(x) - p_N(x) = \frac{f^{(N+1)}(\xi)}{(N+1)!}\mathcal{N}_{N+1}(x)$$

Доказательство.
Для $x = x_i$ теорема выполняется, так как $f(x_i) - p_N(x_i) = 0 = \mathcal{N}_{N+1}(x_i)$. Доказываем для остальных.

Заметим, что $f(x) - p_N(x)$ обращается в ноль по крайней мере в точках x_i. В тех же точках в ноль обращается $\mathcal{N}_{N+1}(x)$. Запишем это таким образом:

$$f(x) - p_N(x) = \mathcal{N}_{N+1}(x)r(x)$$

Здесь $r(x) = \frac{f(x)-p_N(x)}{\mathcal{N}_{N+1}(x)}$ непрерывно дифференцируема $N+1$ раз во всех точках, кроме, вероятно, узлов интерполяции.

Рассмотрим $q(\xi) = f(\xi) - p_N(\xi) - \mathcal{N}_{N+1}(\xi)r(x)$, где x – параметр. Эта функция непрерывна и имеет корень по крайней мере в точках $x_0, \ldots, x_N, x$.

По теореме Ролля, у дифференцируемой функции между корнями обязательно найдётся нуль производной. Функция q непрерывно дифференцируема $N+1$ раз и у неё есть $N+2$ корня. Значит, у неё найдётся ξ_0 – корень $N+1$-ой производной между крайними корнями:

$$q^{(N+1)}(\xi_0) = f^{(N+1)}(\xi_0) - (N+1)!r(x) = 0$$

То есть

$$f(x) - p_N(x) = \mathcal{N}_{N+1}(x)r(x) = \frac{f^{(N+1)}(\xi_0)}{(N+1)!}\mathcal{N}_{N+1}(x)$$

□

Замечание.
Отсюда получаем априорную оценку погрешности:

$$|f(x) - p_N(x)| \leqslant \max_{\xi\in[a,b]}\left(\frac{|f^{(N+1)}(\xi(x))|}{(N+1)!}|\mathcal{N}_{N+1}(x)|\right) = \frac{\|f^{(N+1)}\|_C}{(N+1)!}|\mathcal{N}_{N+1}(x)|$$

Здесь $\|\cdot\|$ – норма в пространстве непрерывных функций, которая супремум нормы значения.

Нам осталось оценить $\mathcal{N}_{N+1}(x)$. Оценим сначала для равномерной сетки с шагом h.

Пусть $x \in [x_{k-1}, x_k]$, тогда $|x_0 - x| \leqslant kh$, $|x_1 - x| \leqslant (k-1)h$, ..., $|x_{k-1} - x| \leqslant h$, $|x_k - x| \leqslant h$, $|x_{k+1} - x| \leqslant 2h$, ..., $|x_N - x| \leqslant (N-k+1)h$. Из этого,

$$|f - p_N| \leqslant \|f^{(N+1)}\|_C \frac{1}{\binom{N+1}{k}} h^{N+1} = O(h^{N+1})$$

Последнее равенство подразумевает фиксированный N и уменьшающийся h (и длину интервала, соответственно). Почему бы и нет.

Кроме того, вообще говоря, это выражение зависит от k. Погрешность тем меньше, чем ближе к середине интервала:

- При $k = 1$ получится

$$|f - p_N| \leqslant \|f^{(N+1)} \frac{h^{N+1}}{N+1}$$

- При $k \approx \frac{N}{2}$ получится

$$|f - p_N| \approx \frac{((N/2)!)^2}{N!} \approx \frac{\left(\left(\frac{N}{2e}\right)^{N/2}\right)^2}{\left(\frac{N}{e}\right)^N} = \frac{1}{2^N}$$

Вообще говоря, по краям уменьшение погрешности линейное, а по центру – экспоненциальное.

А хочется более равномерное распределение этой ошибки. Утверждается (без доказательства), что наименьшая возможная погрешность для интерполяции полиномами достигается опять-таки в корнях многочленов Чебышёва (если отрезок отличается от $[-1, 1]$, нужно применить преобразование $x = 0.5(a + b) + 0.5(b - a)t$.

TODO: Картинки с лекции. Там явно видно, что при равномерной интерполяции внутри отрезка погрешность почти нулевая, а по краям большая. При интерполяции Чебышёвым же погрешность более-менее равномерна.

Замечание. Эффект Рунге (Гиббса)
Полиномиальные интерполяции, как получается, плохо ведут себя по норме C (то есть по поточечному супремуму погрешнности).

4. Сплайны

4.1. Мотивация и историческая справка

До сих пор мы аппроксимировали интерполяцией и сравнивали результат по $\|\cdot\|_C$. Только что мы выяснили, что погрешность может быть очень большой. В случае же, если она накладывается на шум в исходных данных, ситуация получается ещё хуже. Альтернативно, можно делать по-другому:

- Аппроксимировать по интегральной норме ("в среднем").
- Воспользоваться методом наименьших квадратов.
- Аппроксимировать кусочно-полиномиальными функциями (сплайнами).

При интерполяции полиномами степень полинома линейно зависит от количества точек. Если точек десятки тысяч, то это уже очень плохо и не очень нужно. Поэтому и возникает идея разбить отрезок на кусочки и на каждом построить полином.

Говорят, изначально сплайны применяли в судостроении. Там строили функцию, которая проходит через заданный набор точек и минимизирует такой вот возникший из физики функционал:

$$E(y) = \int_a^b \frac{(y''(x))^2}{(1 + (y'(x))^2)^{5/2}} dx$$

Вроде как упругая струна, привязанная к набору точек, будет себя везти именно так. Конечно, минимизировать эту штуку жутко неудобно и сейчас такого не используют. Сейчас считают, что y' достаточно маленькая, чтобы на знаменатель можно было забить, а минимизировать квадрат второй производной гораздо приятнее.

4.2. Определение сплайна

***Определение* 4.1.**
Пусть $x_0 < x_1 < \cdots < x_N$ – набор чисел. Пусть S_n^ν – кусочно-полиномиальная функция, являющаяся полиномом p_n^i степени n на каждом интервале $[x_{i-1}, x_i]$. Пусть также $S_n^\nu \in C^{n-\nu}[x_0, x_N]$. Тогда S_n^ν называется сплайном порядка (степени) n дефекта ν. Точки x_i называются узлами сплайна.

Замечание.
В частности, из требования непрерывной дифференцируемости следует

$$\frac{\partial^k p_n^i}{\partial x^k}(x_i) = \frac{\partial^k p_n^{i+1}}{\partial x^k}(x_i)$$

для $1 \leqslant i \leqslant N-1$, $0 \leqslant k \leqslant n - \nu$.

Замечание.
В частности, кусочно-линейная функция является сплайном степени 1 дефекта 1.

Оценим число свободных параметров сплайна.

У нас N промежутков, на каждом полином степени n с $n+1$ коэффициентом, откуда получаем $N(n+1)$ параметр.

Условия накладывают дополнительные ограничения: в $N-1$ точке $n-\nu$ производных должны совпадать и нижняя из них ещё должна быть непрерывна, итого $(N-1)(n-\nu+1)$ условий.

Итак, всего у нас параметров $F = N(n+1) - (N-1)(n-\nu+1) = \nu(N-1) + n + 1$.

4.3. Задача интерполяции

Пусть $\{x'_i, y_i\}_{i=0}^{M}$ – интерполяционная таблица, а мы ищем сплайн $S_n^\nu(x)$ такой, что $S_n^\nu(x'_i) = y_i$.

Вообще говоря, набор x_i не обязан совпадать с набором x'_i, отсюда у нас появляются разные обозначения для узлов интерполяции и узлов сплайна.

Мы ожидаем, что если F совпадает с $M+1$, то решение может существовать. Но доказывать это будем для каждого типа сплайна отдельно. Часто при этом ставят какие-нибудь дополнительные условия на концы отрезка.

4.4. Параболический сплайн S_2^1

Здесь $F = N+2$. Иллюстрация сплайна, у которого узлы не совпадают с узлами интерполяции.

Потребуем $x'_i \in (x_i, x_{i+1})$.

Теорема 4.1 (Без доказательства)**.**
Рассмотрим задачу интерполяции сплайном S_2^1 для таблицы $\{x'_i, y_i\}$, $M = N-1$ с граничными условиями $\alpha_1 S(a) + \beta_1 S'(a) = \gamma_1$, $\alpha_2 S(b) + \beta_2 S'(b) = \gamma_2$ для $\alpha_i^2 + \beta_i^2 \neq 0$. При выполнени $x'_i \in (x_i, x_{i+1})$ такая задача однозначно разрешима.

4.5. Кубический сплайн S_3^1

На этот раз ищем сплайн с узлами в узлах интерполяции. Тут $F = N+3$. У нас, соответственно, выходит $N+1$ условие из таблицы и два граничных.

Примеры граничных условий:

Естественный (натуральный) сплайн: $S''(x_0) = S''(x_N) = 0$

Обобщение натурального сплайна: $S''(x_0) = A$, $S''(x_N) = B$

Периодический сплайн: $S^{(\rho)}(x_0) = S^{(\rho)}(x_N)$, $\rho = 0, 1, 2$

Not-a-knot сплайн: $S^{(3)}(x_1 - 0) = S^{(3)}(x_1 + 0)$, $S^{(3)}(x_{N-1} - 0) = S^{(3)}(x_{N-1} + 0)$

Последний в точности означает, что в соседних с краями точках непрерывной является ещё и третья производная. Обычно библиотеки при построении сплайна по умолчанию строят именно not-a-knot сплайн, если им не указать граничные условия. Ну, какое-то условие всё-таки надо, а это хорошо тем, что дополнительных чисел от пользователя не требует, в отличие, например, от второго.

Следующая теорема, названная свойством минимальной кривизны, рассказывает, что же такого естественного в естественном сплайне.

Теорема 4.2 (Холлидей)**.**
Пусть $\{x_i, y_i\}$ – таблица интерполяции, $a = x_0$, $b = x_N$. Среди всех $f \in C^2[a,b]$ таких, что $f(x_i) = y_i$, $f''(a) = f''(b) = 0$, минимум функционала $\mathcal{F}(f) = \int\limits_a^b (f''(x))^2 dx$ достигается на естественном сплайне S_3^1.

Доказательство.
Пусть S – естественный сплайн, f – произвольная функция, удовлетворяющая требуемым условиям. Покажем, что $\mathcal{F}(f) - \mathcal{F}(S) \geqslant 0$.

$$\mathcal{F}(f) - \mathcal{F}(S) = \int\limits_a^b ((f'')^2 - (S'')^2)dx = \int\limits_a^b (f'' - S'')^2 dx + 2\int\limits_a^b (f''S'' - (S'')^2)dx =: I_1 + 2I_2$$

Очевидно, $I_1 \geqslant 0$. Покажем, что $I_2 = 0$.

$$I_2 = \int\limits_a^b (f'' - S'')S''dx = \sum_{i=1}^N \int\limits_{x_{i-1}}^{x_i} (f'' - S'')S''dx = \sum_{i=1}^N (f' - S')S''|_{x_{i-1}}^{x_i} - \\ - \sum_{i=1}^N \int\limits_{x_{i-1}}^{x_i} (f' - S')S'''dx$$

Первая сумма равна нулю, так как S'' равна нулю на границах, а во внутренних точках f' и S' равны. Во второй сумме же можно заметить, что S_i''' – константа в силу своей кубической природы:

$$I_2 = -\sum_{i=1}^N S_i''' \int\limits_{x_{i-1}}^{x_i} (f' - S')dx = -\sum_{i=1}^N S_i''' (f - S)|_{x_{i-1}}^{x_i} = 0$$

□

4.6. Существование и единственность кубического сплайна

Здесь мы приведём алгоритм, однозначно находящий кубический сплайн. Заодно докажем единственность.

Обозначим $h_i = x_i - x_{i-1}$.

Строим начиная со второй производной. Пусть $M_i = S''(x_i)$ – значения второй производной сплайна в узлах. Вторая производная – непрерывная кусочно-линейная функция. Значит, на $[x_{i-1}, x_i]$ она себя вёдет так:

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

Проинтегрируем:

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

Здесь у нас вылезла при интегрировании константа d_i. Ещё проинтегрируем:

$$S(x) = \frac{M_i}{6h_i}(x - x_{i-1})^3 - \frac{M_{i-1}}{6h_i}(x_i - x)^3 + \frac{d_i}{2}(x - x_{i-1}) - \frac{d_i}{2}(x_i - x) + c_i$$

Здесь мы дополнительную константу $\frac{d_i h_i}{2}$ внесли в вылезшую константу c_i, потому что можем.

Значения в точках интерполяции дают нам ограничения на значения в x_i и x_{i-1}:

$$\begin{aligned} \frac{M_{i-1}}{6}h_i^2 - \frac{d_i}{2}h_i + c_i &= y_{i-1} \\ \frac{M_i}{6}h_i^2 + \frac{d_i}{2}h_i + c_i &= y_i \end{aligned}$$

Эта система линейная относительно c_i и d_i, найдём их:

$$\begin{aligned} c_i &= \frac{y_i + y_{i-1}}{2} - \frac{M_i + M_{i-1}}{12}h_i^2 \\ d_i &= \frac{y_i - y_{i-1}}{h_i} - \frac{M_i - M_{i-1}}{6}h_i \end{aligned}$$

Теперь все кусочные функции мы умеем находить, если зафиксируем M_i. Осталось написать какие-то ограничения на них. Используем непрерывность $S'(x)$ во внутренних узлах:

$$\frac{M_i h_i}{2} + d_i = -\frac{M_i h_{i+1}}{2} + d_{i+1}$$

Используем выражение для d_i:

$$\frac{M_i h_i}{2} + \frac{y_i - y_{i-1}}{h_i} - \frac{M_i - M_{i-1}}{6}h_i = -\frac{M_i h_{i+1}}{2} + \frac{y_{i+1} - y_i}{h_{i+1}} - \frac{M_{i+1} - M_i}{6}h_{i+1}$$

Вынесем константы направо

$$\frac{h_i}{h_i + h_{i+1}}M_{i-1} + 2M_i + \frac{h_{i+1}}{h_i + h_{i+1}} = \frac{6}{h_i + h_{i+1}}\left(\frac{y_{i+1} - y_i}{h_{i+1}} - \frac{y_i - y_{i-1}}{h_i}\right)$$

Это при $0 < i < N$. Если добавить граничное условие, скажем, $M_0 = M_N = 0$, то останется $N - 1$ переменная и на них $N - 1$ условие. Получили СЛАУ.

В матрице СЛАУ почти все коэффциенты нулевые. Ненулевые только три центральные диагонали. На главной диагонали 2, а на двух соседних что-то заведомо меньшее единицы. Соответственно,

$$|d_{ii}| > \sum_{j \neq i}^{N} |d_{ij}|$$

То есть матрица обладает диагональным преобладанием. Позже мы докажем, что она в таком случае невырождена и научимся находить решение СЛАУ за линейное время.

4.7. Базис в пространстве естественных сплайнов

Если нам нужно решить задачу интерполяции на одних и тех же данных много раз, то может оказаться полезным тот факт, что естественные сплайны на одних и тех же сплайнов образуют линейное пространство.

Упражнение.
Естественные сплайны на узлах $\{x_0, \dots, x_N\}$ образуют линейное пространство размерности $N+1$ с базисом

$$\{S_k(x)\}_{k=0}^{N}: \quad S_k(x_i) = \delta_{ik}$$

Любой естественный кубический сплайн единственным образом представим в виде

$$S(x, \{y\}) = \sum_{k=0}^{N} y_k S_k(x)$$

4.8. Неестественные кубические сплайны

Вообще говоря, конечно, у аппроксимируемой функции далеко не всегда $f''(x_0) = f''(x_N) = 0$, и мы рискуем получить большую погрешность в окрестности концов, предполагая, что это верно. Чтобы этого не случалось, поступают так:

- Если вторые производные откуда-то вдруг известны, можно в качестве начальных условий предположить $M_0 = f''(x_0)$, $M_N = f''(x_N)$.
- Если известна первая производная, точнее будет задать её:

$$S'(x_0) = -\frac{1}{3}M_0 h_1 - \frac{1}{6}M_1 h_1 + \frac{y_1 - y_0}{h_1}$$

- Обычно и этого нет. Тогда можно построить интерполирующий полином по точкам x_0, x_1, x_2, x_3 и задать его производную как производную изначальной функции (как в предыдущем варианте).
- Кроме того, нередко берут другие условия, например, строят те же not-a-knot сплайны.

4.9. B-сплайны

Тут было немного рекламы без доказательств, в конспекте пока пусто :(

TODO:

4.10. Аппроксимация в гильбертовом пространстве

До сих пор мы решали задачу интерполяции, то есть приближали функцию другой функцией, гарантируя небольшое расхождение совпадением значений во многих точках. Вообще говоря, можно гарантировать небольшое расхождение по какой-нибудь норме $\|\cdot\|$ явно.

Задачу там решаем ту же: линейной аппроксимации, то есть поиска такой функции $f^* = \sum_{i=0}^{n} c_i^* \varphi_i(x)$, где c_i^* – константы, а φ_i – линейно независимы, что $\|f - f^*\|$ достигает минимума.

Эта задача иногда тоже решается, но для разной нормы алгоритмы получаются разные, часто гораздо сложнее, чем интерполяция.

Пример.
Теорема Вейерштрасса говорит, что любую гладкую функцию можно сколь угодно точно приблизить по норме $\|\cdot\|_C$ полиномами. Однако мы уже выяснили, что в общем случае степень полиномов нужна большая.

Однако есть простой приятный случай: это Гильбертово пространство. Пусть у нас задано скалярное произведение $\langle\cdot,\cdot\rangle$ и норма $\|f\|^2 = \langle f, f\rangle$. Тогда утверждается, что задача имеет единственное решение. Более того, тогда $f - f^*$ ортогональна всем φ_i, а коэффициенты c_i^* определяются из системы уравнений

$$\sum_{i=0}^{N}\langle\varphi_i, \varphi_k\rangle c_i^* = \langle f, \varphi_k\rangle$$

Если же ещё и все φ_i попарно ортогональны, то получается приятная формула

$$c_i^* = \frac{\langle f, \varphi_i\rangle}{\langle\varphi_i,\rangle\varphi_i}$$

Такие системы нам уже встречались и ещё будут встречаться.

- Разложение в ряд Фурье
- Метод наименьших квадратов
- Разложение по ортогональным полиномам (ещё будет у нас)
  Тут используется скалярное произведение

$$\langle f, g\rangle = \int_a^b f(x)g(x)\rho(x)dx, \quad \rho(x) \geqslant 0$$

5. Дифференцирование

5.1. Введение

По определению, производная – это предел $\frac{\Delta f}{\Delta x}$. Но считать этот предел численно было бы не слишком эффективно.

Во-первых, если функция задана с шумом, то при вычислении функцию с маленьким Δx можно получить совершенно не то значение, на которое мы рассчитываем при взятии производной.

Во-вторых, $|f|$ может быть достаточно большим, а тогда Δf будет разностью больших по модулю близких чисел и при вычислении возникнет большая погрешность.

Вообще, правило большого пальца: если **можно** обойтись без численного вычисления производной, это нужно сделать, потому как совсем хорошо эта задача не решается. Но, конечно, бывают ситуации, когда обойтись нельзя. Например, при решении дифференциальных уравнений, не сводящихся к интегральным.

Численное интегрирование, напротив, шума в данных не слишком боится. Возникает любопытная ситуация, когда аналитически проще считать производную, а вычислительно проще брать интегралы.

5.2. Дифференцирование интерполяционного полинома

Первый способ дифференцирования, который мы рассмотрим – это построить какую-нибудь интерполяцию функции в каких-нибудь точках (полиномами или сплайном) и объявить, что производная нашей функции похожа на производную этой интерполяции. Ну а посчитать производную у полинома или сплайна мы справимся. Оценим погрешность дифференцирования интерполяционного полинома.

Вспомним, что

$$f(x) - p(x) = \frac{f^{(N+1)}(\xi)}{(N+1)!} \prod_{i=0}^{N} (x - x_i)$$

Отсюда возникает естественная оценка

$$|f^{(n)}(x) - p^{(n)}(x)| \leqslant C \frac{\|f_C^{(N+1)}}{(N+1-n)!} \max_i |x - x_i|^{N+1-n}$$

Однако пользоваться такой формулой прямо не слишком удобно.

Мы помним, что интерполяционный полином единственный. В конкретной задаче нам удобнее пользоваться его аналитечиским представлением в форме Ньютона:

$$p(x) = \sum_{k=0}^{N} [x_0, \ldots, x_k] f \prod_{i=0}^{k-1} (x - x_i)$$

Возьмём n-тую производную. Все члены, степени которых меньше n, умрут, n-тый член суммы станет константой, $n+1$-ый станет линейной функцией, и так далее. Несложно убедиться, что произведения превратятся в суммы всех произведений нужной степени:

$$p^{(n)}(x) = n! \left\{ [x_0, \dots, x_n]f + [x_0, \dots, x_{n+1}]f \sum_{i=0}^{n}(x - x_i) + [x_0, \dots, x_{n+1}]f \sum_{j>i\geqslant 0}^{j=n+1} (x - x_i)(x - x_j) + \dots \right\}$$

А теперь оставим только первое слагаемое.

$$f^{(n)}(x) \approx n! \left([x_0, \dots, x_n]f + O(\sum_{i=0}^{n}(x - x_i)) \right)$$

Пусть $h = \max |x - x_i|$. Тогда такая аппроксимация будет иметь порядок (h). А ещё можно заметить, что в точке $x^* = \frac{1}{n+1} \sum x_i$ второй член обратится в ноль, поэтому там порядок точности будет $O(h^2)$.

Если оставить два слагаемых, то для нахождения точки x^* надо решать квадратное уравнение. Однако в этой точке порядок будет $O(h^3)$. А для k слагаемых в специальных точках получим порядок $O(h^{k+1})$. Это явление называется сверхсходимостью.

А ещё нужно сказать, что так-то мы не x^* подбираем по x_i, а вовсе наоборот, строим интерполяцию в таких точках, чтобы x^* совпал с точкой, где нам, собственно, нужно посчитать производную. А тут уже можно строить эти точки конструктивно.

TODO: Я так понял, что если взять точки с равным шагом, то x^* окажется посередине, а докажем мы это позже.

5.3. Конечные разности

Тут было много рукомахательства в доказательствах, объяснялось это нежеланием углубляться в матановый формализм, так как курс, в общем-то, не об этом. С практической точки зрения это означает, что в этом разделе будет несколько плохо определённых понятий, про которые примерно понятно, что они означают.

Мотивация: получить выражение для производной поприятнее.

***Определение* 5.1** (Конечные разности)**.**
Пусть $f(x) \in C$, $h = \Delta x$.

Конечная разность первого порядка определяется так:

$$\Delta_h f = \Delta f(x) = f(x + h) - f(x)$$

(символ h иногда опускается, если он, например, всегда один и тот же).

Конечная разность высшего порядка определяется рекурсивно.

$$\Delta^n_{h_1 h_2 \dots h_n} f = \Delta_{h_n} \left(\Delta^{n-1}_{h_1 h_2 \dots h_{n-1}} f \right)$$

Теорема 5.1 (Свойства конечных разностей)**.**

1. К.Р. не зависят от порядка сдвигов:

$$\Delta^n_{h_1 \dots h_n} f = \Delta^n_{h_{\sigma_1} \dots h_{\sigma_n}} f$$

2. $\Delta^{N+1}_{h_1 \dots h_N} p_N(x) = N! a_N h_1 \dots h_N = const$

3. $\Delta^{N+1}_{h_1 \dots h_{N+1}} p_N(x) = 0$
4. Пусть все интервалы одинаковы, $\Delta^k = \Delta^k_{hh\dots h}$. Тогда

$$k![x_0, \dots, x_k]f = \frac{\Delta^k f(x_0)}{h^k}$$

   Обобщение на неравномерный шаг неверно.

5. $\Delta(\alpha f + \beta g) = \alpha \Delta f + \beta \Delta g$
6. $\Delta^k(\Delta^l f) = \Delta^{k+l} f = \Delta^l(\Delta^k f)$.

Про доказательство: про первое и шестое свойство не сказано ничего, но вроде бы они нигде и не использовались позже. Третье очевидным образом следует из второго. Про четвёртое была дана схема доказательства – индукция по N. Пятое доказывается очевидной индукцией.

Про второе свойство доказательство описывается так: при вычислении выражения вида $(x+h)^n - x^n$ степень на единицу понижается, поэтому в конце останется константа. Множитель a_N остаётся при этом снаружи за скобками, а от выражения внутри старший член (остальные всё равно сократятся) останется равным nhx^{n-1}.

Вообще эти объекты являются дискретным аналогом прозводных и для них даже есть похожие формулы. Но записываются они гораздо сложнее~~, а мы тут руками машем, а не формулы записываем~~, а нам в рамках данного курса достаточно интуитивного понимания.

Далее мы работаем со всякими операторами, делающими функции из функций. Это, например, Δ^k. Или $\frac{d}{dx}$. Мы будем выполнять с ними арифметические действия и применять к ним какие-то функции и понятно, что это будет значить.

Теорема 5.2 (Связь производной с конечной разностью)**.**

$$\frac{d}{dx} = \frac{1}{h}\ln(1+\Delta)$$

Доказательство.
Будем предполагать, что f – аналитическая.

$$(1+\Delta)f = f(x+h) - \sum_{n=0}^{\infty} \frac{1}{n!}\left(h\frac{d}{dx}\right)^n f(x) = \exp\left(h\frac{d}{dx}\right) f(x)$$

А теперь выразим отсюда $\frac{d}{dx}$.

$$\frac{d}{dx} = \frac{\ln(1+\Delta)}{h}$$

□

Замечание.
По модулю аналитичности f это было строгое доказательство, если об этом хорошо подумать.

Замечание.

$$\frac{d}{dx} = \frac{\ln(1+\Delta)}{h} = \frac{1}{h}\left(\Delta - \frac{\Delta^2}{2} + \frac{\Delta^3}{3} + \dots\right)$$

Пример.
Односторонняя формула второго порядка:

$$\frac{df}{dx} \approx \frac{1}{h}\left(\Delta - \frac{\Delta^2}{2}\right) = \frac{1}{h}\left(-\frac{3}{2}f(x) + 2f(x+h) - \frac{1}{2}f(x+2h)\right)$$

Одностороняя – потому что для неё достаточно значений справа от x. Используется для аппроксимации краевых условий.

Чтобы сделать из этого производную старшего порядка, надо повторить эту операцию:

$$\frac{d^2}{dx^2}f = \frac{1}{h}\ln(1+\Delta)\ln(1+\Delta)f$$

5.4. Вычислительная погрешность формул дифференцирования

Формула для вычисления производной через конечные разности первого порядка по сути у нас и есть та формула, которую мы в начале главы осуждали:

$$f'(x) \approx \frac{1}{h}\left(f(x+h) - f(x)\right)$$

Оценим её погрешность. Во-первых, есть погрешность из самой формулы $O(h)$:

$$R_1 \leqslant M_1 h$$

А ещё значения $f(x_i)$ могут быть известны с погрешностью $|\varepsilon_i| \leqslant E$. В реальном мире так будет (почти) всегда, как минимум из-за существования машинной точности:

$$R_2 = -\frac{\varepsilon_1 - \varepsilon_0}{h}, \quad |R_2| \leqslant \frac{2E}{h}$$

Итого:

$$|R| \leqslant |R_1| + |R_2| \leqslant g(h) = M_1 h + \frac{2E}{h}$$

Это значение достигает минимума на $h_0 = \sqrt{\frac{2E}{M_1}}$, а $g(h_0) = \sqrt{8EM_1}$. Ни при каком h погрешность не будет лучше $O(\sqrt{E})$. Для машинной точности это означает половину верных разрядов.

Чуть получше получается для формул высшего порядка – для порядка k получится $O(E^{\frac{k}{k+1}})$.

6. Интегрирование

Численно считать значение интегралов человечество умеет как раз довольно-таки неплохо.

Зачем это нужно?

- Многие интегралы не имеют аналитического представления.
- Иногда хоть аналитическое выражение и существует, но гораздо проще берётся численно.
- В проекционных методах. Там много интегралов. Как пример проекционного метода – у нас уже была интерполяция. Хотя там интегралов не было.
- Решение дифференциальных/интегральных уравнений.

Методы, которые мы здесь рассмотрим:

- Методы, основанные на интерполяции.
- Составные формулы.
- Адаптивные методы.
- Формулы со свободными узлами.
- Формулы для специальных весов.

6.1. Постановка задачи

Методы, которые мы изучаем, будут укладываться примерно в следующую схему, называемую *квадратурной формулой*:

$$\int_a^b f(x)\rho(x)dx \approx \sum_{i=1}^{n} w_i f(x_i) \qquad (*)$$

Здесь $\rho(x)$ – вес (обычно $\rho(x) \geqslant 0$). Иногда он всегда равен 1. В некоторых методах он может быть другим. Но в любом случае будем предполагать у него существование моментов ($k \geqslant 0$):

$$\mu_k = \int_a^b x^k \rho(x)dx$$

В квадратурной формуле есть два подбираемых параметра – узлы x_i и веса w_i.

***Определение* 6.1.**
Алгебраической степенью сложности квадратурной формулы называется максимальная степень полинома, для которого $(*)$ выполняется точно.

6.2. Квадратурные формулы Ньютона-Котеса

Здесь $\rho \equiv 1$. Основная идея – замена f на её полиномиальную интерполяцию.

Используем представление полинома в форме Лагранжа:

$$f(x) = \sum_{i=1}^{N} f(x_i)\mathcal{L}_i^N(x) + r_N(x)$$

Обозначим

$$\lambda_i = \int_a^b \prod_{k \neq i} \frac{x - x_k}{x_i - x_k} dx$$

И получим из этого квадратурную формулу Ньютона-Котеса:

$$\int_a^b f(x)dx = \sum_{i=1}^{n} \lambda_i f(x_i) + \int_a^b r_N(x)dx$$

Замечание.
Вспомним замечание к теореме 3.3 и получим оценку погрешности:

$$|R_N[f]| := \int_a^b r_N(x)dx \leqslant \frac{\|f^{(N)}\|_C}{N!} \int_a^b |\mathcal{N}_N(x)|dx$$

Замечание.
В числителе предыдущего замечания видим N-тую производную. Значит, для полиномов степени $\leqslant N-1$ формула верна точно. Отсюда, алгебраическая степень сложности у формулы – хотя бы $N-1$.

А ещё это правда потому, что интерполяционный полином в силу своей единственности совпадёт с исходной функцией.

Далее рассмотрим частные случаи $n = 1, 2, 3$.

Формула прямоугольников
Здесь просто $n = 1$, интерполяционный полином нулевой степени.

Формула левых прямоугольников: $x_1 = a$

$$I[f] \approx (b-a)f(a)$$

Формула правых прямоугольников: $x_1 = b$

$$I[f] \approx (b-a)f(b)$$

Формула (средних) (прямоугольников): $x_1 = (a+b)/2$

$$I[f] \approx (b-a)f\left(\frac{a+b}{2}\right)$$

Вообще, любое непустое подмножество слов “средних” и “прямоугольников” употребляется для обозначения последней формулы.

Погрешность оценивается так:

$$|R_1[f]| \leqslant C_1 \max_{[a,b]} |f'(x)|(b-a)^2$$

Здесь $C_1(b-a)^2$ берётся из интеграла $\mathcal{N}_1(x)$. Для левых и правых прямоугольников $C_1 = \frac{1}{2}$, для средних $\frac{1}{4}$.

На самом деле, средние ещё на порядок лучше, но это мы обсудим позже.

Формула трапеций

Здесь $n = 2$, $x_1 = a$, $x_2 = b$.

Квадратурная формула:

$$I[f] = \int_a^b f(x)dx \approx \frac{b-a}{2}(f(a) + f(b))$$

А погрешность из формулы выше получается

$$|R_2[f]| \leqslant \max_{[a;b]} |f''(x)| \frac{(b-a)^3}{12}$$

Формула Симпсона

Здесь $n = 3$, $x_1 = a$, $x_2 = \frac{a+b}{2}$, $x_3 = b$. Проводим параболу через три точки и считаем её интеграл.

Для удобства перейдём к отрезку $[-1, 1]$ ($y_1 = -1$, $y_2 = 0$, $y_3 = 1$):

$$q(y) = f\left(\frac{a+b}{2} + y\frac{b-a)}{2}\right)$$

$$\int_a^b f(x)dx = \frac{b-a}{2}\int_{-1}^1 q(y)dy$$

В таком случае интерполяционный полином выглядит так:

$$p_2(y) = q(-1)\frac{y(y-1)}{2} - q(0)(y+1)(y-1) + q(1)\frac{y(y+1)}{2}$$

Интегрируя выписанные полиномы $\mathcal{L}_1$, $\mathcal{L}_2$ и $\mathcal{L}_3$, получаем

$$\int_{-1}^1 q(y)dy \approx \frac{q(-1)}{3} + \frac{4q(0)}{3} + \frac{q(1)}{3}$$

Возвращаясь к исходному интервалу, получаем квадратурную формулу Симпсона:

$$\int_a^b f(x)dx \approx \frac{b-a}{6}\left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right)$$

Погрешность в таком случае из формулы выше *по полиному 3-ей степени* оценится как

$$|R_S[f]| \leqslant \max_{[a;b]} |f^{(3)}(x)| \frac{(b-a)^4}{192}$$

Неожиданный факт: интеграл от x^3 по формуле Симпсона посчитается также точно. На самом деле, это следует из его симметричности отночительно нуля (так и так там будет 0). А это означает дополнительный порядок точности: у нас обнулится в остатке ещё один член ряда. А значит, можно оценить погрешность ещё и по полиному 4-ой степени (как бы считам, что в середине у нас кратный узел; это не является строго определённым и доказанным утверждением):

$$|R_S[f]| \leqslant \max_{[a;b]} |f^{(4)}(x)| \frac{(b-a)^5}{2880}$$

TODO: Лекция