Численные методы

Василий Алфёров по лекциям Евгения Яревского

27 января 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. Аппроксимация. Сплайны.

TODO: