

Лекция 4. БПФ, окна, разрешение, усреднение — подробный разбор

Подробное объяснение «на пальцах»: зачем нужна лекция, что она решает, откуда берутся формулы и что означают картинки. Материал рассчитан и на подготовку преподавателя, и на самостоятельное чтение студентом: от первых понятий до практических рецептов и разбора ошибок.

О чём этот текст и как им пользоваться

- Это **подробный конспект** лекции 4: смысл без «тяжёлого» вывода формул.
 - Кому полезен:
 - студенту — чтобы понять, а не заучить;
 - преподавателю — чтобы было чем наполнить рассказ;
 - практику — чтобы знать, какие ручки крутить.
 - Структура:
 - разделы 1–2 — что такое спектр и почему считать «в лоб» дорого;
 - раздел 3 — как устроено БПФ;
 - разделы 4–6 — конечная запись, окна, разрешение, миф про zero-padding;
 - разделы 7–11 — практика: Уэлч, PSD, точная частота, спектрограмма, пример;
 - разделы 12–21 — шпаргалка, ошибки, код, рецепты, задания, план.
 - Формат — списки: так удобнее готовиться, повторять и объяснять.
 - Числовые примеры взяты из ноутбука `lecture04.ipynb`.
-

Цели лекции (что должен уметь студент)

1. Объяснить, что такое **БПФ**:
 - это не новое преобразование, а **быстрый алгоритм** для ДПФ;
 - результат точно совпадает с ДПФ;
 - выигрыш — в тысячи и десятки тысяч раз.
2. Показать идею **«разделяй и властвуй»**:
 - одно большое ДПФ → два маленьких;
 - рекурсия → $\log_2 N$ уровней.
3. Понять **«бабочку»** и **бит-реверс**.
4. Объяснить, откуда берётся **утечка**.
5. Осознанно выбирать **окно** под задачу.
6. Понимать $\Delta f = 1/T$ и почему **zero-padding не улучшает разрешение**.
7. Пользоваться **методом Уэлча** и читать **PSD**.
8. Строить **спектрограмму**.
9. Сделать всё это в **Python**.

Зачем каждый пункт на практике:

- БПФ — чтобы считать спектр в реальном времени.
- Разрешение — чтобы различать близкие частоты (дефект vs помеха).
- Окна — чтобы видеть слабые линии рядом с сильными.
- Уэлч и PSD — чтобы измерять шум, а не «мохнатый» шумный спектр.
- Спектрограмма — чтобы работать с меняющимися сигналами.

1. Напоминание: что такое спектр и зачем он нужен

1.1 Сигнал и его «радуга»

- **Сигнал** — величина, меняющаяся во времени:
 - напряжение на датчике;
 - вибрация подшипника;
 - звук микрофона;
 - ток двигателя.
- Во времени сигнал — извилистая кривая, где «всё перемешано».
- **Спектр** — разложение сигнала на простые синусоиды с их амплитудами и фазами.
- Аналогия: белый свет через **призму** → цвета; спектр — «радуга сигнала».
- «Призма» здесь — арифметика: пробуем синусоиды и смотрим, какие совпали.
- Практический смысл:
 - превратить «непонятную кривую» в несколько чисел;
 - найти дефект, невидимый глазом;
 - принять решение: проблема есть/нет, на какой частоте, растёт ли она.

1.2 Формула ДПФ словами

- Есть N отсчётов: $x[0]$, $x[1]$, ..., $x[N-1]$.
- ДПФ даёт N комплексных чисел $X[0]$, $X[1]$, ..., $X[N-1]$.
- Смысл $X[k] = \sum x[n] \cdot e^{-j \cdot 2\pi \cdot k \cdot n/N}$:
 - берём пробную синусоиду частоты номер k;
 - умножаем сигнал на неё и складываем;
 - совпали частоты → слагаемые складываются в большое число;
 - не совпали → гасят друг друга, почти ноль.
- ДПФ — это **корреляция** сигнала с набором синусоид.
- Пик появляется там, где произошло совпадение.

1.3 Амплитуда, фаза, мощность

- $|X[k]|$ — **амплитуда** (насколько «громко» звучит частота).
- $\arg X[k]$ — **фаза** (с каким сдвигом присутствует).
- $|X[k]|^2$ — **мощность** составляющей.
- Для анализа чаще смотрят амплитуду или мощность.
- Фаза нужна для:
 - измерения задержек и дальности;
 - фазовой локализации (RFID, радар);
 - когерентного накопления и синхронизации.

1.4 Вещественный и комплексный сигнал

- **Вещественный** сигнал (напряжение, звук) даёт **две** сопряжённые линии: $+f$ и $-f$.
- Поэтому для вещественного сигнала достаточно **половины** спектра (`rfft`).
- **Комплексный (аналитический)** сигнал даёт **одну** линию на $+f$.
- Аналитический сигнал удобен в связи (I/Q).

- Отрицательные частоты — удобство математики; для вещественного сигнала они симметричны положительным.

1.5 Ось частот и бины

- Реальная частота: $f_k = k \cdot f_d / N$, где f_d — частота дискретизации.
- Сетка этих частот — **бины**; шаг сетки $\Delta f = f_d / N$.
- Между бинами спектр не определён «напрямую» — там интерполяция.
- Ось строят функциями `fftfreq` (двусторонняя) и `rfftfreq` (односторонняя).

1.6 Маленький пример

- $f_d = 8000$ Гц, $N = 8 \rightarrow \Delta f = 1000$ Гц; бины 0, 1000, ..., 7000 Гц.
- Косинус ровно 1000 Гц → попадёт в бин, две чёткие линии ± 1000 Гц.
- Косинус 1250 Гц → **между** бинами, «размазанный» холмик (утечка).

1.7 Как выглядит спектр типичных сигналов

- **Чистый тон**: одна-две узкие линии.
- **Периодический сигнал** (меандр, треугольник): набор гармоник кратных основной.
- **Шум**: размазанная «полка» по всем частотам.
- **Вибрация**: несколько линий (вращение, дефекты) плюс шум.
- **Речь/музыка**: множество линий, меняющихся во времени.
- **Радиосигнал**: несущая и боковые полосы (зависит от модуляции).
- Понимание «типового вида» помогает сразу заметить аномалию.

1.8 Краткая история БПФ

- Идею деления на чётные/нечётные знал ещё **Гаусс** (начало XIX века).
- Широко известной она стала в 1965 году — статья **Cooley–Tukey**.
- Причина популярности — компьютеры и потребность в быстрой обработке.
- Сегодня БПФ — в основе аудио, связи, изображений, радаров, МРТ, сейсмоки.

2. Проблема №1: считать спектр «в лоб» дорого

2.1 Сколько это операций

- Один коэффициент $X[k]$ — это N произведений и сложений.
- Коэффициентов $N \rightarrow$ итого N^2 операций (квадратичная сложность $O(N^2)$).

2.2 Почему это критично

- Секунда аудио при 44.1 кГц: $N = 44\,100$, $N^2 \approx 1.9 \cdot 10^9$.
- Реальное время: пересчитывать десятки раз в секунду.
- Длинные записи и изображения: N ещё больше.
- Время растёт как квадрат длины записи — быстро «упирается».

2.3 Сравнение в числах

N	Прямое ДПФ N^2	БПФ $N \cdot \log_2 N$	Во сколько раз меньше
64	$\sim 4 \cdot 10^3$	$\sim 4 \cdot 10^2$	~ 16
1024	$\sim 10^6$	$\sim 10^4$	~ 100
65 536	$\sim 4 \cdot 10^9$	$\sim 10^6$	~ 4000
10^6	$\sim 10^{12}$	$\sim 2 \cdot 10^7$	$\sim 50\,000$

- Вывод: выигрыш растёт с N и становится **астрономическим**.
- Поэтому БПФ — один из важнейших алгоритмов XX века.

2.4 Что такое «сложность $O(\cdot)$ » на пальцах

- $O(N^2)$ — время растёт как квадрат длины: удвоили N — время вчетверо.
- $O(N \cdot \log_2 N)$ — почти линейно: удвоили N — время чуть больше, чем вдвое.
- На больших N разница между кривыми определяет, «успеем или нет».
- В реальном времени это разница между «работает» и «не работает».

2.5 Как ускоряли до БПФ (интуиция)

- Заметили, что одни и те же множители встречаются много раз.
- Их можно переиспользовать, а не считать заново.
- Отсюда идея делить задачу пополам и переиспользовать результаты.

3. БПФ: быстрое преобразование Фурье

3.1 Главная мысль

- **БПФ** — не приближение и не другое преобразование.
- Оно считает **тот же** ДПФ, но быстрее: $O(N \cdot \log_2 N)$ вместо $O(N^2)$.
- Результат совпадает с точностью машинной арифметики.
- «Быстрое» относится к скорости вычислений, а не к упрощению результата.

3.2 «Разделяй и властвуй»

- Пусть N чётно.
- Делим отсчёты на **чётные** и **нечётные**:
 - чётные: $x[0], x[2], x[4], \dots$;
 - нечётные: $x[1], x[3], x[5], \dots$
- Спектр всего сигнала выражается через два спектра половинной длины:
 - $E[k]$ — спектр чётных;
 - $O[k]$ — спектр нечётных;
 - и поворотный множитель W .
- Причина: частота в показателе «пересчитывается» для половинной длины ($W_{N^2} = W_{(N/2)}$).
- Далее — **рекурсия**: каждую половину снова делим пополам.
- Пока не останутся ДПФ длины 1–2.

- Глубина дерева — $\log_2 N$; отсюда множитель $\log_2 N$.
- Уравнение сложности: $T(N) = 2 \cdot T(N/2) + O(N)$.
- Его решение — $O(N \cdot \log_2 N)$.

3.3 Почему деление «работает»

- Поворотные множители обладают симметрией:
 - $W^{(k+N/2)} = -W^k$;
 - $W^{(2k)} = W_{(N/2)}^k$.
- Именно эта симметрия позволяет соединять половины «почти бесплатно».
- Без неё деление не давало бы выигрыша.

3.4 «Бабочка»

- Две половины соединяются **«бабочкой»**:
 - $X[k] = E[k] + W \cdot O[k]$;
 - $X[k + N/2] = E[k] - W \cdot O[k]$.
- Одно и то же $W \cdot O[k]$ **прибавляется и вычитается**.
- За одну операцию — сразу **две** точки спектра.
- Стоимость: одно комплексное умножение и два сложения.
- Название — из-за формы схемы (два входа, два выхода, перекрестье).

3.5 Ручной пример для $N = 4$

- Вход $x = [x_0, x_1, x_2, x_3]$.
- Разделяем: чётные $[x_0, x_2]$, нечётные $[x_1, x_3]$.
- ДПФ длины 2:
 - $E[0] = x_0 + x_2, E[1] = x_0 - x_2$;
 - $O[0] = x_1 + x_3, O[1] = x_1 - x_3$.
- Точки спектра:
 - $X[0] = E[0] + W^0 \cdot O[0]$;
 - $X[2] = E[0] - W^0 \cdot O[0]$;
 - $X[1] = E[1] + W^1 \cdot O[1]$;
 - $X[3] = E[1] - W^1 \cdot O[1]$.
- Здесь $W = e^{(-j \cdot 2\pi/4)}$.
- Вместо 16 умножений (в лоб) — два умножения на W .
- Для $N = 8$ экономия ещё заметнее.

3.6 Ручной пример для $N = 8$ (идея)

- Делим на чётные (0, 2, 4, 6) и нечётные (1, 3, 5, 7).
- Каждую четвёрку — снова на чётные/нечётные.
- Получаем четыре ДПФ длины 2.
- Соединяем «бабочками» в три стадии.
- Число «дорогих» умножений резко падает.

3.7 Бит-реверс

- Плата за деление: входные отсчёты идут в **бит-реверсивном** порядке.

- Правило: номер в двоичном виде читается **справа налево**.
- Пример: $1 = 001_2 \rightarrow 100_2 = 4$.
- Для $N = 8$: 0, 4, 2, 6, 1, 5, 3, 7.
- На выходе порядок естественный: $X[0..7]$.
- В коде — одна перестановка массива перед БПФ.

3.8 Как выглядит граф 8-точечного БПФ

- Три вертикальные **стадии**.
- На каждой — четыре **бабочки**.
- Слева — вход в бит-реверсивном порядке.
- Справа — спектр $X[0..7]$ в обычном порядке.
- Структура регулярная $\rightarrow$ удобно для процессора и железа (FPGA, ASIC).

3.9 Итеративный и рекурсивный варианты

- **Рекурсивный** БПФ — ближе к определению, но с накладными расходами.
- **Итеративный** — по стадиям, «на месте» (in-place); так делают библиотеки.
- In-place экономит память: результат пишется поверх входа.
- Порядок стадий обеспечивает бит-реверс заранее.

3.10 Варианты алгоритма и практика

- **radix-2** — деление пополам; требует $N = 2^m$.
- **radix-4** — меньше операций.
- **split-radix** — ещё быстрее.
- **mixed-radix** — для составных N .
- **Простое большое N** — смешанные алгоритмы или округление до степени двойки.
- **Вещественный сигнал** — `rfft`, вдвое дешевле.
- **Библиотеки**: FFTW, `pocketfft` (NumPy/SciPy), MKL, `cuFFT` (GPU).
- Правило: по возможности брать N **степенью двойки**.

3.11 Где БПФ применяется

- Аудио: сжатие, эквалайзеры, распознавание речи.
- Связь: OFDM (Wi-Fi, LTE, DVB), фильтрация.
- Медицина: МРТ, ЭЭГ, ЭКГ.
- Техника: вибродиагностика, радары, сейсмика, астрономия.

4. Проблема №2: конечная запись и утечка

4.1 Что мы на самом деле считаем

- Записываем **кусоч** сигнала длиной T секунд.
- Формально это умножение на **прямоугольное окно** (1 внутри записи, 0 снаружи).
- Правило: **умножение во времени = свёртка в спектре**.
- Спектр прямоугольного окна — «холмик с хвостами» (`sinc`).
- Значит, чистый тон **размазывается**.

- Это и есть **утечка** — следствие конечности записи, а не ошибка.

4.2 Свёртка «на пальцах»

- Свёртка — «размазывание» одной функции другой.
- Аналогия: резкий звук в комнате с эхом «размазывается» во времени.
- Здесь по частоте: резкую линию «размазывает» хвост окна.
- Чем «чище» спектр окна, тем меньше размазывание.

4.3 Откуда боковые лепестки (sinc)

- Спектр прямоугольного окна:
 - **главный лепесток** — центральный пик;
 - **боковые лепестки** — затухающие «волны» по бокам.
- Главный лепесток → **разрешение** (шире — хуже).
- Боковые лепестки → **маскировка** слабых линий.
- У прямоугольного окна боковые высокие (–13 дБ) — плохо для слабых сигналов.

4.4 Численный пример утечки

- $f_{\text{д}} = 8000 \text{ Гц}$, $N = 256 \rightarrow \Delta f = 31.25 \text{ Гц}$.
- Тон ровно в бине (например, 1000 Гц) → одна чёткая линия.
- Тон на полбина (1015.6 Гц) → энергия растекается по соседним бином.
- С прямоугольным окном боковые падают медленно.
- С окном Hann — падают быстро.

4.5 Окна: лекарство от утечки

- Обрезаем сигнал **мягко**: окно равно 1 в центре и плавно идёт к 0 у краёв.
- Аналогия: мягкая диафрагма фотоаппарата даёт меньше «звёздочек».
- Эффект: боковые лепестки падают.
- Цена: главный лепесток **расширяется** — разрешение чуть хуже.
- Это **неустрашимый компромисс**.

4.6 Популярные окна

Окно	Главный лепесток	Боковые лепестки	Когда брать
Прямоугольное (boxcar)	2 бина	–13 дБ	максимальная резкость, чистые тона
Hann	4 бина	–31 дБ	«рабочая лошадка», общий анализ
Hamming	4 бина	–43 дБ	чуть больше динамического диапазона
Blackman-Harris	8 бинов	–92 дБ	слабая линия рядом с сильной

Окно	Главный лепесток	Боковые лепестки	Когда брать
flat-top	10 бинов	-93 дБ	точная амплитуда (калибровка)

- **Прямоугольное**: ничего не меняем; хорошо, если частоты точно в бинах.
- **Hann**: плавный «колокол»; выбор по умолчанию.
- **Hamming**: похоже на Hann, но не доходит до нуля; чуть ниже боковые.
- **Blackman-Harris**: очень низкие боковые; шире главный лепесток.
- **flat-top**: широкий плоский верх; точная амплитуда, плохое разрешение.

4.7 «Паспорт» окна: coherent gain и ENBW

- **Coherent gain** — среднее значение окна:
 - поправка на амплитуду линии;
 - Hann ≈ 0.5 , Hamming ≈ 0.54 , flat-top ≈ 0.22 .
- **ENBW** — эквивалентная шумовая полоса:
 - поправка на шум и PSD;
 - прямоугольное 1.0, Hann 1.5, Hamming 1.36, flat-top 3.77.
- Без учёта этих коэффициентов нельзя сравнивать спектры с разными окнами.
- `scipy.signal.welch` учитывает их автоматически.

4.8 Как выбрать окно

- Разделённые тона, важна резкость → **прямоугольное** или **Hann**.
- Слабая линия рядом с сильной → **Hann** или **Blackman-Harris**.
- Точная амплитуда → **flat-top**.
- Не знаете что взять → **Hann**.

4.9 Частые вопросы про окна

- «Окно убирает шум?» — нет, окно против утечки; шум убирает усреднение.
- «Окно искажает сигнал?» — ослабляет края; при нормировке амплитуда верна.
- «Можно без окна?» — да, если частоты точно в бинах и сигнал чистый.
- «Почему пик ниже?» — это coherent gain.
- «Почему пик шире?» — расширенный главный лепесток.

5. Разрешение по частоте

5.1 Главная формула

- $\Delta f = f_{\text{д}} / N$, а так как $T = N / f_{\text{д}}$, то $\Delta f = 1 / T$.
- Чем **дольше** запись, тем **тоньше** разрешение.
- Δf — минимальное расстояние, различимое по построению.

5.2 Диапазон и разрешение — разные вещи

- **Частота дискретизации** задаёт **диапазон** (до $f_{\text{д}}/2$, частота Найквиста).

- **Длительность записи** задаёт **разрешение** Δf .
- Поднять $f_{\text{д}}$ → шире диапазон, но разрешение (при том же T) не улучшится.
- Аналогия: диапазон — «какие ноты слышим»; разрешение — «какие близкие ноты различаем».

5.3 Число Найквиста

- Верхняя частота: $f_{\text{д}} / 2$.
- Всё выше — «заворачивается» (алиасинг), если нет антиалиасингового фильтра.
- Поэтому перед дискретизацией ставят фильтр.

5.4 Критерий Рэлея

- Тона различимы, если расстояние между ними **больше** Δf .
- Для N_{fft} нужно примерно $\approx 2\Delta f$.
- То есть окно немного «съедает» разрешение, но даёт динамику.

5.5 Опыт с двумя тонами

- Тона 1000 и 1040 Гц.
- $N = 128 \rightarrow \Delta f = 62.5 \text{ Гц} \rightarrow$ сливаются.
- $N = 1024 \rightarrow \Delta f \approx 7.8 \text{ Гц} \rightarrow$ два пика.
- Вывод: «раздвинуть» близкие частоты можно **удлинением записи**.

5.6 От чего ещё зависит разрешение

- Длительность записи T (главное).
- Форма/ширина окна.
- Отношение сигнал/шум.
- Наличие мешающих линий (боковые лепестки).

5.7 Примеры Δf для разных T

Длительность T	$\Delta f = 1/T$
0.1 с	10 Гц
0.5 с	2 Гц
1 с	1 Гц
10 с	0.1 Гц

- Хотите различить тона в 1 Гц — нужна запись около секунды и больше.

6. Миф: zero-padding

6.1 Что делают

- Дополняют сигнал **нулями** до большей длины перед ДПФ.
- Получают «гладкий» спектр с плотной сеткой частот.

6.2 Почему это не улучшает разрешение

- Новых данных не появляется — мы не смотрели дольше.
- Это **интерполяция** той же спектральной кривой.
- Слипшиеся тона сольются и на плотной сетке.

6.3 Когда zero-padding полезен

- Красивый гладкий график.
- Точное положение пика (для интерполяции).
- Приведение длины к степени двойки.
- Линейная свёртка: дополнение до $N \geq N_a + N_b - 1$.
- Итог: удобство, а не разрешение.

6.4 Как не спутать

- Если пики стали «чаще» — это интерполяция, а не новое разрешение.
 - Если пики стали «уже и раздельнее» — это уже удлинение записи.
-

7. Усреднение: метод Уэлча

7.1 Почему один спектр «мохнатый»

- Одна периодограмма $|X|^2$ **неустойчива**; разброс не падает с ростом N .
- Причина: одна «реализация» случайного процесса.
- Аналогия: одно измерение шума случайно; сто усреднённых — устойчивы.

7.2 Идея по шагам

1. Резать запись на **сегменты**.
2. Брать их с **перекрытием**.
3. Применять **окно**.
4. Считать **БПФ** и $|\cdot|^2$.
5. **Усреднять** по сегментам.

7.3 Что получаем и чем платим

- Получаем: несмещённую оценку с меньшим разбросом.
- Разброс падает как $\sqrt{K}$ (K — число сегментов).
- Платим: разрешение определяется длиной **сегмента**.

7.4 Как выбирать параметры

- `nperseg` (длина сегмента):
 - больше → лучше разрешение, больше разброс;
 - меньше → меньше шум, хуже разрешение.
- `overlap` (перекрытие): обычно 50% (`nperseg // 2`).
- `nperseg` — желательно степень двойки.

- Разрешение: $f_{\text{д}} / \text{nperseg}$ должно быть достаточно мелким.
- Больше сегментов → глаже кривая, но больше вычислений.

7.5 Сколько нужно усреднений

- $K = 1$ → как одна периодограмма (шумно).
- $K = 10$ → заметно глаже.
- $K = 100$ → устойчивая кривая.
- Разброс $\approx 1/\sqrt{K}$: чтобы улучшить в 2 раза, нужно в 4 раза больше сегментов.

7.6 Родственники метода

- **Периодограмма** — один сегмент (шумная).
- **Бартлетт** — сегменты без перекрытия.
- **Уэлч** — перекрытие + окно (стандарт).
- **Медианное усреднение** — устойчиво к выбросам.
- **Мультитейпер** — несколько ортогональных окон.

7.7 Порядок действий

1. Определить нужное Δf .
 2. По нему выбрать длину сегмента.
 3. Взять окно (Hann).
 4. Перекрытие 50%.
 5. Посчитать `welch`.
 6. Убедиться: линии на месте, шум «успокоился».
-

8. PSD: спектральная плотность мощности

8.1 Что это

- Для шума «амплитуда на частоте» бессмысленна.
- Правильная мера — **мощность на герц** (PSD).
- **Площадь под PSD = полная мощность** сигнала.

8.2 Две нормировки

- `scaling='density'` → «мощность²/Гц» (для шумов и широкополосных сигналов).
- `scaling='spectrum'` → «мощность² в бине» (для дискретных линий).

8.3 Формула с ENBW

- $\text{PSD} = |W|^2 / (f_{\text{д}} \cdot \text{ENBW})$.
- Благодаря ENBW уровень шумовой полки не зависит от окна.
- В `welch` это уже учтено.

8.4 Мощность в полосе

- Мощность в полосе $[f_1, f_2] \approx$ сумма PSD по этой полосе, умноженная на шаг Δf .
- Отсюда: чем шире полоса, тем больше «собранная» мощность.

8.5 Зачем децибелы

- Динамический диапазон спектра огромен.
- В дБ он «сжимается» и читается глазом: $10 \cdot \log_{10}(\text{PSD})$.
- Таблица для ориентира:

Отношение мощностей	В дБ
$\times 2$	+3
$\times 10$	+10
$\times 100$	+20
$\times 1000$	+30
$\times 10^6$	+60

- Минус дБ: мощности нельзя складывать «на глаз».

8.6 Типичные ошибки с PSD

- Смешивать density и spectrum на одном графике.
- Сравнивать разные окна без нормировки.
- Забывать про Δf при переходе к мощности в полосе.
- Считать уровень шума не зависящим от разрешения (он зависит от полосы).

9. Точная оценка частоты

- Сетка даёт частоту с шагом Δf ; истина может быть между бинами.
- Приём: **параболическая интерполяция** по трём точкам вокруг пика.
- Берём три значения логарифма спектра (левое, центр, правое).
- Формула даёт поправку δ (от -0.5 до $+0.5$ бина).
- Уточнённая частота: $f_o \approx (k + \delta) \cdot \Delta f$.
- Пример из ноутбука: истина 100.4 Гц, сетка ≈ 3.9 Гц, оценка — 100.34 Гц.
- Более точные методы:
 - интерполяция по двум окнам;
 - оценка по фазе;
 - Прони, MUSIC — для близких тонов.
- Для вибрации и радиосигналов обычно достаточно интерполяции.

10. Спектрограмма: сигнал во времени и по частоте

10.1 Что такое STFT

- Обычный спектр предполагает, что сигнал всё время одинаков.

- **STFT**: спектр в **скользящем окне**.
- Результат — **спектрограмма**: время (X), частота (Y), мощность (цвет).

10.2 Что видно на спектрограмме

- **Чирп** (частота растёт) — наклонная линия.
- **Короткий всплеск** — вертикальное пятно.
- **Долгий тон** — горизонтальная линия.
- **Дрейф частоты** — изгиб линии.
- Одним обычным спектром этого не увидеть.

10.3 Компромисс время–частота

- **Короткое окно** → хорошо «когда», плохо «какая частота».
- **Длинное окно** → точно по частоте, но размыто по времени.
- Это принцип неопределённости, а не недоработка метода.

10.4 Когда спектрограмма нужна

- Нестационарные сигналы (речь, музыка, разгон).
- Поиск коротких событий (щелчки, удары, разряды).
- Отслеживание изменения частоты.
- Когда обычный спектр «смешивает» разные режимы.

10.5 Родственные инструменты

- **Вейвлет-преобразование** — адаптивное окно; лучше для широкополосных всплесков.
- **Спектрограмма** — простой и наглядный инструмент на каждый день.
- **Мультирейперная спектрограмма** — сглаженнее.

11. Реальный пример: вибрация, шаг за шагом

- Данные: вибрация агрегата.
- Составляющие: база 50 Гц, дефект 120 Гц, слабая гармоника 240 Гц, шум.
- Шаг 1. Одна периодограмма: шум «мохнатый», линия 240 Гц тонет.
- Шаг 2. Окно Hann + Уэлч: шум сглажен, линия 240 Гц различима.
- Шаг 3. PSD в дБ: удобный масштаб, видно превышение над полкой.
- Шаг 4. Уточнение частот интерполяцией.
- Итог: не «картинка», а **основа для решения**.

12. Шпаргалка (словарь терминов)

Термин	Что означает
ДПФ	N чисел → N чисел спектра
БПФ	быстрый алгоритм того же ДПФ, $O(N \log N)$

Термин	Что означает
Бабочка	операция соединения двух половин спектра
Бит-реверс	перестановка входа БПФ
Окно	плавное обрезание записи
Утечка	размазывание линии из-за конечности записи
Главный лепесток	ширина пика → разрешение
Боковые лепестки	«хвосты», мешают видеть слабые линии
Coherent gain	поправка на амплитуду
ENBW	поправка на шум (эквивалентная шумовая полоса)
Разрешение	$\Delta f = 1/T$
Zero-padding	интерполяция, а не разрешение
Периодограмма	квадрат модуля БПФ одной записи
Метод Уэлча	усреднение периодограмм по сегментам
PSD	мощность на герц
Спектрограмма	спектр во времени (STFT)
Бин	узел частотной сетки ДПФ
Найквист	верхняя частота $f_{д}/2$

13. Частые ошибки и заблуждения

- «Подниму $f_{д}$ — улучшу разрешение» → нет, разрешение задаёт T .
- «Дополню нулями — увижу детали» → это интерполяция.
- «Окно убирает шум» → окно против утечки, шум убирает усреднение.
- «БПФ приближённое» → результат точно совпадает с ДПФ.
- «Разные окна сравнимы по высоте пиков» → только с coherent gain.
- «Больше точек всегда лучше» → разрешение = время, устойчивость = усреднение.
- «Утечка — баг» → закономерность конечной записи.
- «Спектрограмма = спектр» → спектрограмма добавляет ось времени.

14. Типичные ошибки в коде (scipy)

- Применили окно **дважды**: сначала вручную, потом внутри `welch`.
- Передали `fs` в герцах, а частоты трактовали как бины (или наоборот).
- Использовали `fft` вместо `rfft` для вещественного сигнала (лишняя работа).
- Забыли `fftshift` там, где нужен «центрированный» спектр.
- Взяли `nperseg` больше длины сигнала.
- Сравнили `spectrum` и `density` без пересчёта.
- Не учли coherent gain при измерении амплитуды вручную.
- Путали `overlap` в отсчётах и в процентах.
- Строили спектрограмму без `scaling`, а потом удивлялись уровню.
- Смотрели амплитуду сигнала до вычета среднего (постоянная составляющая).

15. Практические рецепты по задачам

- **Аудио (речь, музыка):**
 - `perseg 1024–4096`, Hann, перекрытие 50%;
 - спектрограмма с окном 512–2048 для событий.
 - **Вибрация / диагностика:**
 - `perseg 1024–8192`, Hann;
 - ищем гармоники и боковые полосы дефектов.
 - **Радио / связь:**
 - окно Hann или Blackman–Harris для слабых сигналов;
 - I/Q, аналитический сигнал.
 - **ЭКГ / биосигналы:**
 - `perseg` небольшой (0.5–2 с) для нестационарности;
 - окно Hann; вычитать постоянную составляющую.
 - **Измерение частоты:**
 - интерполяция пика; при возможности — фазовый метод.
-

16. Мини-лаборатория (что сделать в ноутбуке руками)

1. Сгенерировать чистый тон ровно в бине и посмотреть ДПФ.
 2. Сдвинуть частоту на полбина и увидеть утечку.
 3. Применить окно Hann и сравнить боковые лепестки в дБ.
 4. Сравнить спектры при $N = 128$ и $N = 1024$ (два близких тона).
 5. Дополнить нулями и убедиться, что разрешение не улучшилось.
 6. Реализовать простой БПФ radix-2 и сверить с `numpy.fft`.
 7. Посчитать `periodogram` и `welch`, сравнить «мохнатость».
 8. Построить PSD в `density` и `spectrum`, объяснить разницу.
 9. Сделать спектрограмму chirp и короткого всплеска.
 10. Оценить частоту параболической интерполяцией.
-

17. План рассказа на 2 академических часа

1. **5 мин.** ДПФ и его цена $O(N^2)$.
 2. **15 мин.** БПФ: разделяй и властвуй, бабочка, бит-реверс, сложность.
 3. **15 мин.** Конечная запись → утечка; окна; компромисс лепестков.
 4. **15 мин.** Разрешение $\Delta f = 1/T$; два тона; миф про zero-padding.
 5. **15 мин.** Уэлч и PSD: зачем усреднять; `density` vs `spectrum`.
 6. **15 мин.** Спектрограмма; компромисс время–частота; точная частота.
 7. **10 мин.** Реальный кейс (вибрация), рецепт `scipy.signal`, задание №4.
- **Приём:** сначала плохая картинка, потом — что делает приём.
 - Давать «до/после»: утечка → окно, шум → Уэлч, короткая запись → длинная.
-

18. Вопросы для самопроверки

1. Чем БПФ отличается от ДПФ по результату и стоимости?
 2. Зачем на входе БПФ бит-реверс?
 3. Что физически означает утечка?
 4. Почему окно расширяет главный лепесток?
 5. Как выбрать окно для слабой линии рядом с сильной?
 6. Почему $\Delta f = 1/T$, а не связано с f_d ?
 7. Почему zero-padding не улучшает разрешение?
 8. Зачем перекрытие сегментов в Уэлче?
 9. Чем density отличается от spectrum?
 10. В чём компромисс время–частота?
-

19. Связь с другими темами курса

- **Лекция 2** — непрерывное преобразование Фурье и спектры (идея).
 - **Лекция 3** — ДПФ (что считаем).
 - **Лекция 4** — БПФ и практический спектральный анализ (как считать).
 - **Лекция 5** — модуляция и I/Q (радиосигналы).
 - **Лекции 6+** — фильтры, Z-преобразование, многомерная обработка.
-

20. Итог одной страницей

- ДПФ — точный спектр, но $O(N^2)$.
 - БПФ — тот же результат за $O(N \cdot \log N)$.
 - Вход БПФ — бит-реверс; соединение — «бабочки».
 - Конечная запись → утечка; окна → компромисс «разрешение/динамика».
 - Разрешение $\Delta f = 1/T$; zero-padding — только интерполяция.
 - Уэлч → устойчивая PSD; спектрограмма → время–частота.
 - Практика — `scipy.signal (get_window, rfft, welch, spectrogram)`.
-

21. Главная мысль и что дальше

- Спектр — не картинка, а **измерение**.
- Чтобы ему верить, мало уметь считать быстро (БПФ).
- Нужно понимать искажения:
 - конечная запись (утечка);
 - выбор окна (разрешение против динамики);
 - шум (дисперсия оценки).
- И уметь отделять реальные линии от артефактов:
 - окна;
 - усреднение (Уэлч);
 - нормировка PSD.
- **Дальше (лекция 5):** спектры радиосигналов и модуляция — АМ/ЧМ/ФМ, цифровые созвездия, комплексная огибающая (I/Q).

Приложения

22. Математика за формулами — словами

22.1 Почему формула ДПФ именно такая

- Ядро $e^{(-j \cdot 2\pi \cdot k \cdot n/N)}$ — это «пробная» синусоида, повернутая на комплексной плоскости.
- Умножение и суммирование — это мера похожести сигнала на эту синусоиду.
- Если сигнал содержит эту частоту, произведение даёт постоянную составляющую, которая при суммировании накапливается.
- Если не содержит — произведение колеблется и в сумме даёт почти ноль.
- Поэтому ДПФ — это корреляция; она «выпячивает» присутствующие частоты.

22.2 Формула Эйлера — словами

- $e^{(j\theta)} = \cos\theta + j \cdot \sin\theta$.
- Геометрический смысл: точка на **единичной окружности** под углом θ .
- Мнимая единица j — поворот на 90° .
- Отсюда комплексная экспонента — удобный способ «крутить» фазу.
- Множитель $e^{(-j \cdot 2\pi \cdot k \cdot n/N)}$ — это дискретные повороты на равные углы.
- Для N точек получается N равномерно расставленных «стрелок» — базис ДПФ.

22.3 Умножение во времени = свёртка в спектре

- Возьмём произведение двух сигналов: $z = x \cdot w$.
- Его спектр равен **свёртке** спектров X и W .
- Именно поэтому конечная запись (умножение на окно) «размазывает» спектр.
- Чем уже спектр окна, тем меньше размазывание.
- У прямоугольного окна спектр широкий — потому утечка велика.

22.4 Откуда берётся «заворачивание» и периодичность

- Дискретизация во времени делает спектр **периодическим** по частоте.
- Период повторения — $f_{\text{д}}$.
- Поэтому ДПФ можно «центрировать» (`fftshift`) и смотреть отрицательные частоты.
- Это же свойство объясняет алиасинг: частоты выше $f_{\text{д}}/2$ накладываются.

22.5 Откуда ENBW

- ENBW показывает, во сколько «бинов» окно собирает шум.
- Формула $N \cdot \sum w^2 / (\sum w)^2$ — отношение энергии квадрата окна к квадрату суммы.
- Прямоугольное окно «собирает» ровно один бин ($\text{ENBW} = 1$).
- Более гладкие окна шире — $\text{ENBW} > 1$.
- Через ENBW нормируют шумовую полку, чтобы она не зависела от окна.

22.6 Почему разброс Уэлча падает как $1/\sqrt{K}$

- Каждый сегмент даёт независимую оценку периодограммы.
- Усреднение K независимых оценок уменьшает разброс как $\sqrt{K}$.
- Поэтому дисперсия падает как $1/K$.
- Отсюда правило: чтобы улучшить гладкость вдвое, нужно вчетверо больше сегментов.
- Расплата — разрешение определяется длиной сегмента, а не всей записи.

22.7 Равенство Парсеваля

- Энергия сигнала во времени равна энергии в спектре (с нормировкой).
- Практический смысл: вычисления «не теряют» энергию.
- Полезно для проверки корректности ДПФ/БПФ.

22.8 Частоты и бины

- Бин k соответствует частоте $f_k = k \cdot f_{\text{д}}/N$.
 - Шаг сетки $\Delta f = f_{\text{д}}/N = 1/T$.
 - Число уникальных бинов:
 - N для комплексного сигнала;
 - $N/2 + 1$ для вещественного.
-

23. Вопросы и развёрнутые ответы

1. **Почему БПФ быстрее, если считает то же самое?** Оно переиспользует одни и те же множители за счёт деления задачи пополам; число операций становится $N \cdot \log_2 N$ вместо N^2 .
2. **БПФ даёт другой результат?** Нет. Отличия только из-за машинного округления; математически результат тот же.
3. **Можно ли считать БПФ для любого N ?** Для $N = 2^m$ проще всего (radix-2). Для других N есть mixed-radix и специальные алгоритмы; часто проще добить нулями до степени двойки.
4. **Что делать, если частоты точно не в бинах?** Использовать окно (Hann/Blackman–Harris) и/или интерполировать пик.
5. **Почему у прямоугольного окна боковые только –13 дБ?** Это следствие «резкого» обрыва записи; спектр такого окна — медленно затухающий sinc.
6. **Какое окно универсальное?** Hann: хороший компромисс между разрешением и динамическим диапазоном.
7. **Как увидеть слабую линию рядом с сильной?** Blackman–Harris (или Hann) + усреднение; при необходимости — вычитание сильной линии.
8. **Почему после zero-padding график гладкий, но пики не разделились?** Потому что это интерполяция; новых данных нет.
9. **Что выбрать: разрешение или меньше шума?** Это компромисс: длинный сегмент — лучше разрешение, короткий — меньше шума.

10. **Сколько нужно данных для устойчивого спектра?** Чем больше, тем лучше; ориентир — десятки-сотни сегментов для гладкой PSD.
 11. **Что такое «шумовая полка»?** Средний уровень PSD шума; реальные линии — пики над полкой.
 12. **Чем *density* отличается от *spectrum*?** *density* — мощность на герц (для шума); *spectrum* — мощность в бине (для линий).
 13. **Почему в дБ?** Динамический диапазон спектра огромен; дБ «сжимает» масштаб для глаза.
 14. **Как точно измерить частоту?** Интерполяция пика (парабола по логарифму), фазовый метод, Прони/MUSIC.
 15. **Что такое спектрограмма?** Спектр во времени: STFT в скользящем окне.
 16. **Почему на спектрограмме нельзя быть точным и по времени, и по частоте?** Принцип неопределённости: короткое окно — время, длинное — частота.
 17. **Нужно ли вычитать среднее?** Да, постоянная составляющая (нулевая частота) иначе «давит» масштаб.
 18. **Что делать с алиасингом?** Ставить антиалиасинговый фильтр до дискретизации и следить за $f_{\text{д}}/2$.
 19. **Можно ли анализировать нестационарный сигнал одним спектром?** Можно, но он «смешает» режимы; лучше спектрограмма.
 20. **Где это применяется в курсе?** В сквозном проекте: диагностика по вибрации, акустика, радио (RFID), речь.
-

24. Как читать ноутбук лекции (по шагам)

- **Ячейка «стоимость ДПФ/БПФ»:** измеряет время прямого ДПФ и `pr.fft.fft`.
 - Вывод: разрыв растёт с N .
 - **Ячейка «дерево DIT»:** показывает разбиение 8 отсчётов.
 - **Ячейка «бабочка»:** схема соединения двух половин.
 - **Ячейка «граф БПФ»:** три стадии по четыре бабочки.
 - **Ячейка «бит-реверс»:** таблица перестановки индексов.
 - **Ячейка «окна во времени»:** формы окон.
 - **Ячейка «спектры окон»:** главный и боковые лепестки в дБ.
 - **Ячейка «компромисс окна»:** ширина против боковых.
 - **Ячейка «утечка»:** тон между бинами; `rect` против `Hann`.
 - **Ячейка «разрешение»:** два тона при $N = 128$ и $N = 1024$.
 - **Ячейка «zero-padding»:** гладко, но без нового разрешения.
 - **Ячейка «Уэлч»:** одна периодограмма против усреднённой.
 - **Ячейка «PSD»:** *density* против *spectrum*.
 - **Ячейка «спектрограмма»:** чирп и всплески.
 - **Ячейка «вибрация»:** слабая гармоника 240 Гц.
 - **Ячейка «точная частота»:** параболическая интерполяция.
-

25. Другие инструменты спектрального анализа

- **Вейвлеты:**
 - адаптивное окно (широкое на низких частотах, узкое на высоких);
 - хороши для всплесков и нестационарных сигналов.
 - **Банки фильтров:**
 - набор полосовых фильтров;
 - основа аудиокодеков и слуховых моделей.
 - **Параметрические методы (Прони, MUSIC, ESPRIT):**
 - дают сверхвысокое разрешение для близких тонов;
 - чувствительны к модели и шуму.
 - **Спектральное оценивание через модели (AR/ARMA):**
 - короткие записи, гладкие спектры.
 - **Машинное обучение:**
 - спектрограмма как «картинка» для нейросети;
 - классификация состояний (дефект/норма);
 - важно помнить о физике сигнала.
-

26. Историческая и практическая справка

- Идея быстрого ДПФ — фактически у **Гаусса** (около 1805).
 - Широкая известность — **Cooley-Tukey**, 1965.
 - Появление FFTW (1997) сделало БПФ «заводским» стандартом.
 - Сегодня БПФ встроено в процессоры, GPU, DSP и даже микроконтроллеры.
 - В курсе БПФ — рабочий инструмент: от вибродиагностики до радиоприёма.
 - Практический вывод: не изобретать БПФ, а уметь им пользоваться и понимать ограничения.
-

27. Полный числовой пример БПФ (N = 4, вручную)

- Возьмём $x = [1, 2, 3, 4]$.
- **Прямое ДПФ** (для проверки):
 - $X[0] = 1 + 2 + 3 + 4 = 10$;
 - $X[1] = 1 + 2 \cdot (-j) + 3 \cdot (-1) + 4 \cdot (j) = -2 + 2j$;
 - $X[2] = 1 - 2 + 3 - 4 = -2$;
 - $X[3] = 1 + 2 \cdot (j) + 3 \cdot (-1) + 4 \cdot (-j) = -2 - 2j$.
- **Через БПФ (radix-2):**
 - чётные $[1, 3]$, нечётные $[2, 4]$;
 - $E[0] = 1 + 3 = 4$, $E[1] = 1 - 3 = -2$;
 - $O[0] = 2 + 4 = 6$, $O[1] = 2 - 4 = -2$;
 - множитель $W = e^{-j \cdot \pi/2} = -j$.
- Соединяем «бабочками»:
 - $X[0] = E[0] + O[0] = 10$;
 - $X[2] = E[0] - O[0] = -2$;
 - $X[1] = E[1] + W \cdot O[1] = -2 + 2j$;
 - $X[3] = E[1] - W \cdot O[1] = -2 - 2j$.
- Результаты **совпали** с прямым ДПФ.

- Вывод: БПФ — это организация вычислений, а не приближение.
 - Проверить себя: `np.fft.fft([1, 2, 3, 4])` даёт те же числа.
-

28. Чек-лист практика: как анализировать спектр

1. Определить **диапазон** частот, который интересует.
 2. Выбрать **частоту дискретизации** ($f_{\text{д}} \geq 2 \cdot \text{верхняя частота}$).
 3. Определить нужное **разрешение** Δf и отсюда длительность T .
 4. Решить, **стационарный** ли сигнал (нужна ли спектрограмма).
 5. Выбрать **окно** (по умолчанию Hann).
 6. Для шумного сигнала — **усреднить** (Уэлч), подобрать `nperseg/noverlap`.
 7. Проверить **нормировку** (`density/spectrum, coherent gain`).
 8. Построить спектр в **дБ** и отметить ожидаемые линии.
 9. **Уточнить** частоты пиков интерполяцией.
 10. Сравнить с **физикой** процесса и сделать вывод.
-

29. Англо-русский глоссарий

Английский	Русский
DFT	ДПФ
FFT	БПФ
bin	бин, узел сетки
window	окно
leakage	утечка
main lobe	главный лепесток
side lobe	боковой лепесток
coherent gain	когерентное усиление
ENBW	эквивалентная шумовая полоса
zero padding	дополнение нулями
periodogram	периодограмма
Welch method	метод Уэлча
PSD	спектральная плотность мощности
spectrogram	спектрограмма
resolution	разрешение
alias / aliasing	наложение / алиасинг
bit reversal	бит-реверс
radix-2	основание 2
Nyquist	Найквист
decibel (dB)	децибел

30. Ещё вопросы (продолжение)

1. **Почему N степень двойки удобно?** Radix-2 требует $N = 2^m$, и БПФ работает без «хвостов» и специальных случаев.
 2. **Что будет, если N простое?** Прямое radix-2 не ляжет; нужны смешанные алгоритмы или паддинг.
 3. **Влияет ли форма окна на измерение амплитуды?** Да; учитывайте coherent gain.
 4. **Сколько бинов у вещественного сигнала?** $N/2 + 1$.
 5. **Зачем `rfft`?** Вдвое меньше вычислений и памяти для вещественного сигнала.
 6. **Что такое «центрированный» спектр?** После `fftshift`: нулевая частота в центре.
 7. **Что делать с постоянной составляющей?** Вычесть среднее перед анализом.
 8. **Как выбрать `perseg`?** Под нужное разрешение; обычно степень двойки.
 9. **Что если сигнал не стационарный?** Спектрограмма/вейвлеты.
 10. **Как сравнивать два спектра?** Одинаковые окно, длина, нормировка.
 11. **Что такое утечка «на практике»?** Кажущиеся линии рядом с реальной.
 12. **Почему шум «мохнатый»?** Оценка по одной реализации; нужна статистика.
 13. **Как измерить уровень шума?** По шумовой полке PSD.
 14. **Зачем нужен `overlap`?** Чтобы не терять данные на границах сегментов.
 15. **Что читать дальше?** Оппенгейм/Шафер, Сергиенко; статьи Harris и Welch.
-

31. Что студент должен уметь «на выходе»

- **Понимание:**
 - объяснить БПФ и его отличие от ДПФ;
 - объяснить утечку и роль окна;
 - объяснить $\Delta f = 1/T$;
 - объяснить смысл PSD и Уэлча.
 - **Навыки:**
 - считать спектр в Python (`rfft`, `welch`, `spectrogram`);
 - правильно строить ось частот;
 - выбирать окно и параметры усреднения;
 - измерять частоты и амплитуды;
 - отделять реальные линии от артефактов.
 - **Отношение:**
 - спектр — это измерение, а не «картинка для отчёта»;
 - любые выводы проверять на устойчивость (окна, длина, усреднение).
-

32. Разбор по кейсам сквозного проекта

- **Диагностика насосного агрегата (вибрация):**
 - сигнал: ускорение/скорость, десятки-сотни Гц;
 - что искать: оборотная частота, гармоники, боковые полосы дефектов;
 - окно Hann, `perseg` 1024–8192, PSD;
 - вывод: наличие и рост дефекта.
- **Акустический контроль сварных швов:**
 - сигнал: ультразвук, короткие импульсы;

- что искать: время прихода, амплитуда, затухание;
 - важно: широкополосность, огибающая, время-частотный анализ.
 - **Мониторинг зоны (RFID + LiDAR):**
 - RFID: RSSI и фаза, обратное рассеяние;
 - LiDAR: дальность и угол;
 - что искать: присутствие, перемещение, идентичность;
 - далее: слияние сенсоров (фильтр Калмана).
 - **Речевой интерфейс оператора:**
 - сигнал: 16 кГц, речь;
 - что искать: речь/паузы, команды, шум;
 - спектрограмма, мел-признаки, детектор активности (VAD).
 - **Качество электроэнергии (THD):**
 - сигнал: 50 Гц и гармоники;
 - что искать: коэффициенты гармоник, отклонения;
 - окно и точная синхронизация с периодом сети.
 - **Сейсмоакустический мониторинг:**
 - сигнал: низкие частоты, длинная запись;
 - что искать: события, частотный состав;
 - спектрограмма и усреднение.
-

33. Методические заметки для преподавателя

- **Начинать с «боли»:** показать плохой спектр и спросить, что не так.
 - **Демонстрировать «до/после»:**
 - утечка → окно;
 - шум → Уэлч;
 - короткая запись → длинная.
 - **Давать руками:** пусть студент сам поменяет окно и увидит разницу.
 - **Типичные ошибки студентов:**
 - путают диапазон и разрешение;
 - верят zero-padding;
 - сравнивают спектры с разными окнами;
 - забывают о нормировке;
 - не вычитают постоянную составляющую.
 - **Как проверять понимание:**
 - попросить объяснить, почему пик стал шире/ниже;
 - попросить выбрать окно под задачу и обосновать;
 - попросить отличить реальную линию от артефакта.
 - **Связь с практикой:** каждый приём привязать к реальной задаче проекта.
 - **Оценивание:** ноутбук с графиками + один слайд-вывод с обоснованием выбора.
-

34. «Формула → смысл» — короткая шпаргалка

- $X[k] = \sum x[n] \cdot e^{(-j2\pi kn/N)} \rightarrow$ корреляция с синусоидой частоты k .
- $f_k = k \cdot f_{\text{д}}/N \rightarrow$ реальная частота бина k .
- $\Delta f = f_{\text{д}}/N = 1/T \rightarrow$ разрешение задаёт время записи.

- $W_N = e^{(-j2\pi/N)} \rightarrow$ «шаг» поворота на комплексной плоскости.
 - $X[k] = E[k] + W \cdot O[k] \rightarrow$ «бабочка» соединения половин.
 - $\cos\theta = (e^{(j\theta)} + e^{(-j\theta)})/2 \rightarrow$ у вещественного тона две линии $\pm f$.
 - $y = x * h$ (свёртка) $\rightarrow$ умножение спектров (и наоборот).
 - $PSD = |W|^2 / (f_{\text{д}} \cdot ENBW) \rightarrow$ нормировка мощности на герц.
 - $\delta = \frac{1}{2} \cdot (y_{-1} - y_{+1}) / (y_{-1} - 2y_0 + y_{+1}) \rightarrow$ поправка частоты пика.
 - $O(N^2) \rightarrow O(N \cdot \log_2 N) \rightarrow$ суть выигрыша БПФ.
-

35. Аналогии и метафоры (чтобы объяснить любому)

- **Спектр** — призма/радуга: разложение сложного на простое.
 - **ДПФ** — «подбор частоты»: угадываем, какие ноты спрятаны в аккорде.
 - **БПФ** — тот же подбор, но умнее: не перебираем, а делим пополам.
 - **Бабочка** — «развилка»: одно число идёт вверх и вниз.
 - **Бит-реверс** — «перепутанные карточки», которые раскладывают заранее.
 - **Утечка** — «размазывание»: фильтр с крутыми краями плодит «звёздочки».
 - **Окно** — мягкая диафрагма: убирает «звёздочки», но чуть размывает.
 - **Главный лепесток** — толщина «карандаша», которым рисуем линию.
 - **Боковые лепестки** — «эх» карандаша по бокам.
 - **Zero-padding** — «лупа на той же картинке»: больше точек, но не больше деталей.
 - **Разрешение** — «бинокль и время»: близкое различает наблюдение, а не увеличение.
 - **Уэлч** — «усреднить сто измерений», чтобы успокоить случайность.
 - **PSD** — «мощность на герц»: сколько энергии в каждой «щели» спектра.
 - **Спектрограмма** — «партитура во времени»: видно, когда какая нота звучит.
 - **Порог Найквиста** — «скорость съёмки»: слишком быстрое движение «заворачивается».
-

36. Частые вопросы о дискретизации и Найквисте

- **Что такое частота Найквиста?** Верхняя частота $f_{\text{д}}/2$, которую можно корректно оцифровать без алиасинга.
 - **Почему именно половина?** Чтобы описать синусоиду, нужно минимум два отсчёта на период (один — «вверх», один — «вниз»).
 - **Что происходит выше $f_{\text{д}}/2$?** Частота «заворачивается» и притворяется низкой — это алиасинг.
 - **Как с ним бороться?** Антиалиасинговый фильтр до АЦП.
 - **Можно ли поднять $f_{\text{д}}$ и «улучшить» всё?** Шире диапазон — да; разрешение при том же T — нет.
 - **Сколько отсчётов нужно?** Столько, сколько даёт нужную длительность $T = N/f_{\text{д}}$.
 - **Зачем anti-alias, если можно фильтровать цифрового?** Уже поздно: алиасинг необратим.
 - **Как связано T и число периодов?** Чем больше периодов в записи, тем «резче» линия на спектре.
-

37. Практические значения параметров для железа курса

- ESP32-C3, АЦП 12 бит:

- $f_{\text{д}}$ до десятков кГц;
 - квантование даёт шум; полезно усреднять.
 - **Микрофон I2S (звук):**
 - $f_{\text{д}}$ 16 или 48 кГц;
 - Hann, nperseg 1024–4096, спектрограмма для событий.
 - **LiDAR (сканы):**
 - по сути «сигнал» в пространстве (дальность/угол);
 - фильтрация выбросов, а не частотный анализ в первом приближении.
 - **RFID 800 МГц:**
 - RSSI и фаза во времени;
 - фильтрация и сглаживание, спектр для периодических помех.
 - **Вибрация (акселерометр):**
 - $f_{\text{д}}$ 1–10 кГц;
 - Hann, PSD, усреднение; ищем гармоники.
 - **Питающая сеть (THD):**
 - синхронизация с 50 Гц, окно на целое число периодов.
 - Общее правило: **сначала понять сигнал и частоты**, потом выбирать параметры.
-

38. Что делать, если «не получается» (траблшутинг)

- **Пик не на ожидаемой частоте:**
 - проверьте ось частот (fftfreq/rfftfreq, fs, fftshift);
 - не перепутаны ли бины и герцы;
 - нет ли алиасинга (частота выше $f_{\text{д}}/2$).
- **Спектр «слишком мохнатый»:**
 - это одна периодограмма; примените Уэлч;
 - увеличьте число сегментов (меньше nperseg).
- **Слабые линии не видны:**
 - возьмите окно с низкими боковыми (Hann/Blackman–Harris);
 - усредните; уменьшите постоянную составляющую;
 - проверьте, не «тонут» ли линии в боковых лепестках сильной.
- **Амплитуда «неправильная»:**
 - учтите coherent gain;
 - правильно выберите density/spectrum;
 - проверьте нормировку $|X|/N$ или $2|X|/N$.
- **Две близкие частоты сливаются:**
 - увеличьте длительность записи T (не $f_{\text{д}}$);
 - используйте окно с узким главным лепестком;
 - при необходимости — параметрические методы.
- **Спектрограмма «пустая» или «шумная»:**
 - проверьте scaling и перевод в дБ;
 - подберите длину окна (компромисс время–частота).
- **Результаты не повторяются от запуска к запуску:**
 - так и должно быть для шума; усредняйте;
 - фиксируйте генератор случайных чисел для воспроизводимости.
- **БПФ «медленное»:**
 - берите N степень двойки;

- используйте `rfft` для вещественного сигнала;
 - не считайте БПФ в цикле Python без необходимости.
-

39. Итог

- Лекция отвечает на два вопроса: «как считать **быстро**» (БПФ) и «как считать **правильно**» (окна, разрешение, усреднение).
 - Ключевые опоры:
 - $O(N^2) \rightarrow O(N \log N)$;
 - $\Delta f = 1/T$;
 - компромисс «главный/боковые лепестки»;
 - $1/\sqrt{K}$ у метода Уэлча.
 - Всё это — рабочие инструменты спектрального анализа, а не абстракция.
 - Дальше: лекция 5 — модуляция и I/Q; лекции 6+ — фильтры и системы.
-

40. Как строить выводы (научная добросовестность)

- **Не верить одной картинке:** повторите с другим окном, длиной, усреднением.
- **Фиксировать параметры:** окно, N , f_s , n_{perseg} — записывать в отчёт.
- **Проверять устойчивость:** реальная линия не должна «прыгать» при смене окна.
- **Отличать артефакт от сигнала:**
 - гармоники кратных частот — вероятно, реальные;
 - «частокол» вокруг сильной линии — вероятно, боковые лепестки.
- **Проверять физику:** измеренная частота должна объясняться процессом.
- **Указывать погрешность:** Δf , точность амплитуды, число усреднений.
- **Не подгонять вывод:** если данных мало — сказать, что данных мало.
- **Помнить о цели:** спектр нужен для **решения**, а не для красоты.
- **Связь с проектом:** каждый вывод — это шаг к рекомендации для человека.