📖 Chapter: FFT in NumPy — The Two Facts That Rule Them All

◈ The Two Big Rules

Rule #1 — Number of bins = N

If you do an **N-point FFT**, you get **N bins**:

```
k = 0, 1, ..., N-1
```

- Each bin is like a labeled drawer for one frequency.
- More samples → more bins.

Rule #2 — Bin spacing = Δf = fs / N

This is the distance in Hz between bin centers.

- Smaller Δf → better ability to separate close frequencies.
- Also:

$$T_{\text{block}} = \frac{N}{f_s}$$

  and

$$\Delta f = \frac{1}{T_{\text{block}}}$$

  Meaning: longer observation → finer detail.

📊 Quick Table

Parameter	What happens if you increase it?	What stays the same?
N	More bins, smaller Δf, longer observation time	Nyquist limit stays same (fs/2)
fs	Larger Nyquist range, bigger Δf (coarser bins)	Number of bins (if N fixed)

🔍 Case Study — Same fs, Different N

We fix:

- Sampling rate: **fs = 8 Hz**
- Tone: **f0 = 2.5 Hz**
- Compare **N = 8** vs **N = 32**

Case A: N = 8

- Δf = 8 / 8 = **1 Hz**
- Bin centers: [0, 1, 2, 3, 4, -3, -2, -1] Hz
- Observation time: 8 / 8 = **1 s**

Tone 2.5 Hz → between 2 Hz and 3 Hz bins → **spreads energy (spectral leakage)**.

Case B: N = 32

- Δf = 8 / 32 = **0.25 Hz**
- Bin centers include: ..., 2.25, **2.5**, 2.75, ... Hz
- Observation time: 32 / 8 = **4 s**

Tone 2.5 Hz → lands exactly on bin #10 → **sharp single-bin peak**.

💡 **Intuition:**
Longer recording → more cycles observed → can tell 2.50 Hz from 2.49 Hz.

Exercise: Run This and See

```
import numpy as np

fs = 8.0
f0 = 2.5

for N in (8, 32):
    t = np.arange(N) / fs
    x = np.sin(2*np.pi*f0*t)
    X = np.fft.fft(x)
    freqs_pos = np.fft.rfftfreq(N, 1/fs)
    mag_pos = np.abs(X[:len(freqs_pos)]) / N
    if len(mag_pos) > 2:
        mag_pos[1:-1] *= 2
    print(f"N={N}, Δf={fs/N} Hz")
    print("Freq bins:", freqs_pos)
    print("Magnitudes:", np.round(mag_pos, 4))
    print("Peak:", freqs_pos[np.argmax(mag_pos)], "Hz\n")
```

Prediction before running:

- N=8 → smeared peak near 2 or 3 Hz
- N=32 → sharp peak exactly at 2.5 Hz

🎯 Takeaways

1. **N controls resolution** — bigger N → finer Δf.
2. **fs controls coverage** — bigger fs → can see higher frequencies but bins get coarser (if N fixed).
3. **Time vs Frequency trade-off** — better Δf means slower reaction to changes.

🏋 Practice Challenges

1. If fs = 1000 Hz and N = 500, what's Δf?
2. If you double N, what happens to Δf and T?
3. With fs fixed, can you change Nyquist limit by changing N? Why or why not?
4. Write a script to detect the exact bin index for f0 = 60 Hz when fs = 512 Hz and

N = 1024.