Chapter 1 — Foundations of Discrete Time and Sampling

How continuous signals become sequences of numbers and why time steps matter

---

1. The Core Idea

In DSP, a **continuous-time signal** $x(t)$ can't be stored or processed directly — we sample it at fixed intervals to create a **discrete-time sequence** $x[n]$.

---

2. Key Ingredients

- **Sampling frequency** f_s → number of samples per second.
- **Sampling interval** $T_s = 1/f_s$ → time gap between consecutive samples.
- **n** → sample index (integer), maps to time $t_n = nT_s$.

---

3. The First Step in DSP

You are basically taking **snapshots** of the signal's value at regular intervals. Each snapshot is stored as a number (could be integer, float, complex).

---

4. Why It Matters

- Too low f_s → aliasing (fake frequencies appear).
- Correct f_s → preserves the original waveform's information.
- Time steps are **uniform** → essential for using FFT, convolution, etc.

Chapter 2 — Foundations of Complex Numbers and Euler's Bridge

How one formula links circles, waves, cosines, and sines

---

1. The Core Idea

Euler's formula:

$$e^{j\theta} = \cos(\theta) + j\sin(\theta)$$

It says a **complex exponential** is both a **rotating point** on a circle and the combination of cosine (real) and sine (imaginary) components.

---

2. Why This Is a Big Deal in DSP

- Every periodic waveform can be expressed as a sum of these rotating phasors.
- Multiplying by $e^{j\theta}$ rotates a signal in **phase space**.
- Cosine = projection on **real axis**, sine = projection on **imag axis**.

---

3. The Unit Circle View

- Radius = 1
- Angle = θ (in radians)
- Going around the circle → time increases.
- **One full revolution** = 2π radians = 360°.

---

4. The Frequency Link

If θ changes at a constant rate, the speed of rotation = frequency.

$$\theta[n] = 2\pi f \cdot t_n$$

Chapter 3 — Foundations of Phasor Rotation and Projection in Time & Frequency

How advancing in time and angle on the unit circle reveals the cosine and sine components of a signal

---

(this is exactly the one I gave you earlier, with the time-step + phase-step breakdown and code example)
