Pre-lab Question 2

Your Ronchi ruling consists of a series of long parallel slits (slit width b, period a). Monochromatic light incident on the object diffracts and forms a far-field (Fraunhofer) diffraction pattern.

(a) Derive Eq. (32) for the far-field diffraction intensity pattern of N slits.

(b) Make sample plots for different values of N and a, and describe qualitatively how each parameter affects:

- the spacing between diffraction orders,
- the width of diffraction peaks,
- and the overall envelope of the pattern.

(c) Explore the effect of varying the duty cycle b/a. How does it affect the relative intensity of diffraction orders (including which orders may be suppressed)?

(d) In the experiment, diffraction orders will not be perfectly sharp and may broaden with order. Briefly explain at least two physical reasons why a real diffraction pattern may deviate from the ideal prediction of Eq. (32). (Hint: consider finite source size, finite spectral bandwidth, and imperfect alignment/aberrations.)

Eq. (32):

$$I(\theta) = I_0\left(\frac{\sin\beta}{\beta}\right)^2\left(\frac{\sin(N\alpha)}{N\sin(\alpha)}\right)^2$$

Where:

$$\alpha = \frac{ka\sin\theta}{2}, \quad \beta = \frac{kb\sin\theta}{2}, \quad k = \frac{2\pi}{\lambda}$$

B)

Make sample plots for different values of N and a, and describe qualitatively how each parameter affects: • the spacing between diffraction orders, • the width of diffraction peaks, • and the overall envelope of the pattern.

$$I(\theta) \propto \left(\frac{\sin(N\alpha)}{N\sin(\alpha)}\right)^2$$

In [9]:
```
import matplotlib.pyplot as plt
plt.style.use('dark_background')
import numpy as np
```

Note:

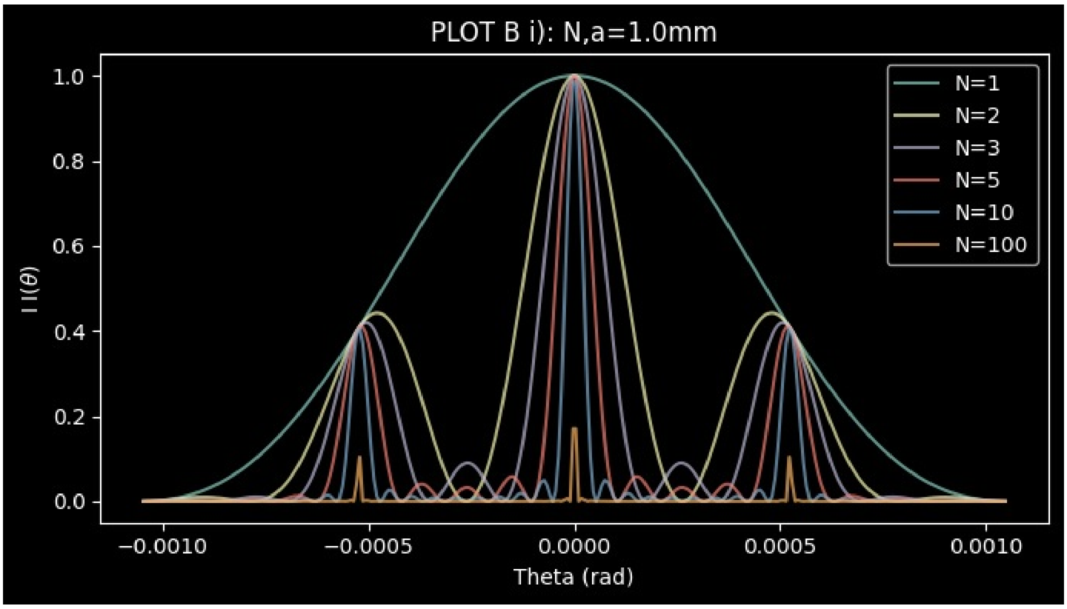
Here a is the grating period of the Ronchi ruling and b is the slit width (close to b = a/2, but it may differ slightly). (LabScript: Pg 11)

In [10]:
```
def I(theta, N, a, lam=None, I0=1.0):
    """
    theta : array of angles (rad)
    N     : number of slits
    a     : grating period (m)
    lam   : wavelength (m), default 525 nm
    I0    : overall I scale
    """

    lam = (lam or 525) # nm (Lab Script Pg 22)
    lam = lam * 10**(-9) # m
    b = a/2 #  (Lab Script Pg 11 and inspired by code in Pg 12)

    k = 2*np.pi / lam
    alpha = 0.5 * k * a * np.sin(theta)
    beta  = 0.5 * k * b * np.sin(theta)

    single_slit = (np.sin(beta) / beta)**2
    interference = (np.sin(N*alpha) / (N*np.sin(alpha)))**2

    return I0 * single_slit * interference
```

i) Keeping **a** constant, Changing N

In [11]:

```python
theta = np.linspace(-np.pi/3000,np.pi/3000,300)
N_lst = [1,2,3,5,10,100] # different numbers of slits
a_lst = [1,10,100,1000] # mm

# Keeping a constant, Changing N
#plt.subplot(1,2,1)
plt.figure(figsize=(7,4))
for i in range(len(N_lst)):
    a = a_lst[0]
    a = a * 10**(-3) # m
    #plt.subplot(len(N_lst),2,i+1)
    # sharex = True
    plt.plot(theta, I(theta,N=N_lst[i],a=a) , label=f"N={N_lst[i]}",alpha=0.7)

plt.title(f"PLOT B i): N,a={a*1000}mm")
plt.xlabel("Theta (rad)")
plt.ylabel(r"I I$(\theta)$")
plt.legend()
plt.tight_layout()
plt.show()
```

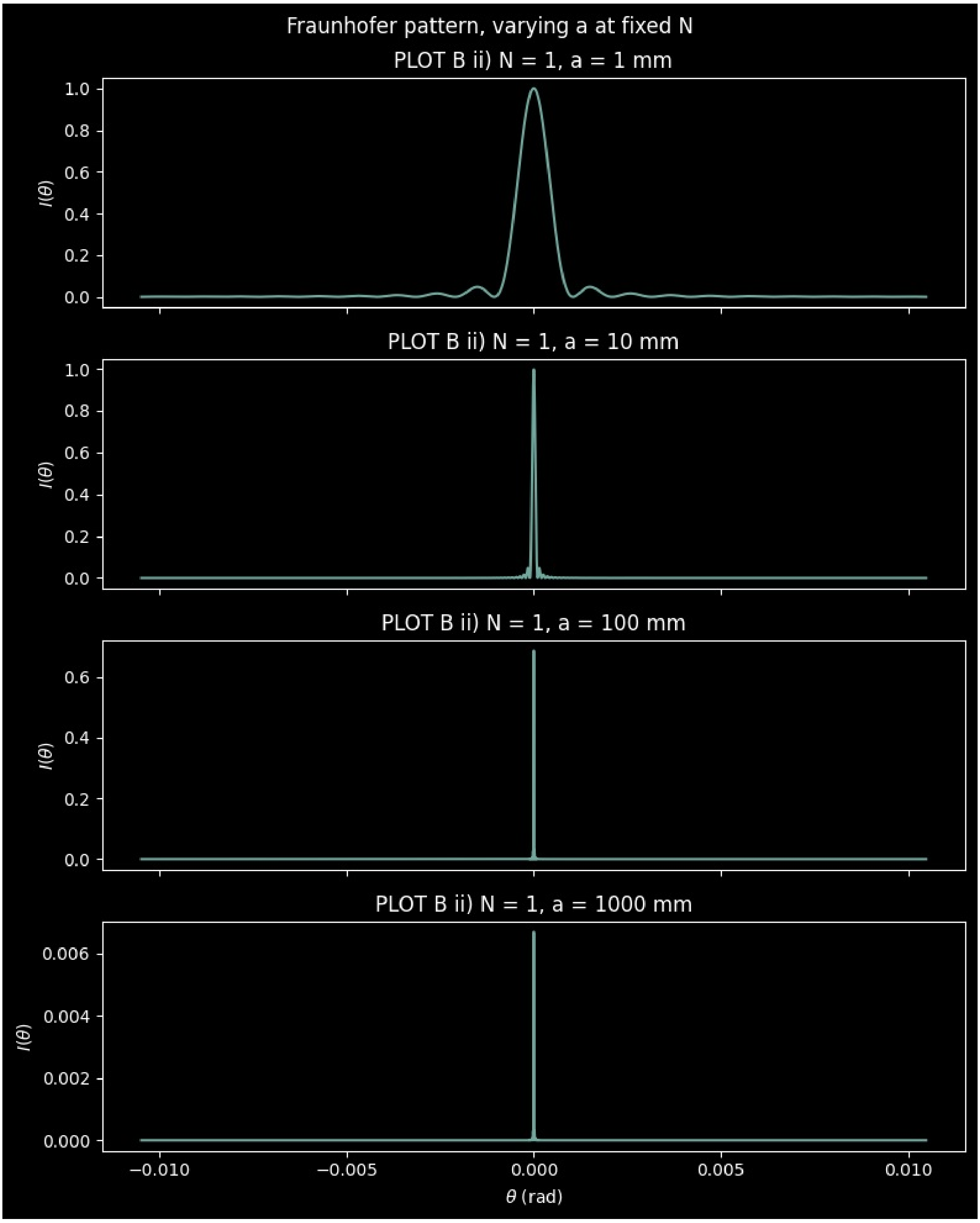


ii) Vary **a** at fixed **N**

In [12]:

```python
theta = np.linspace(-np.pi/300,np.pi/300,3000)
plt.figure(figsize=(16,8))

N = N_lst[0]

fig, axs = plt.subplots(len(a_lst), 1, figsize=(8, 10), sharex=True)

for i, a_mm in enumerate(a_lst):
    a = a_mm * 1e-3  # convert mm -> m
    I_vals = I(theta, N=N, a=a)

    axs[i].plot(theta, I_vals, alpha=0.8)
    axs[i].set_ylabel(r"$I(\theta)$")
    axs[i].set_title(f"PLOT B ii) N = {N}, a = {a_mm} mm")

axs[-1].set_xlabel(r"$\theta$ (rad)")
fig.suptitle("Fraunhofer pattern, varying a at fixed N", y=0.98)
fig.tight_layout()
plt.show()
```

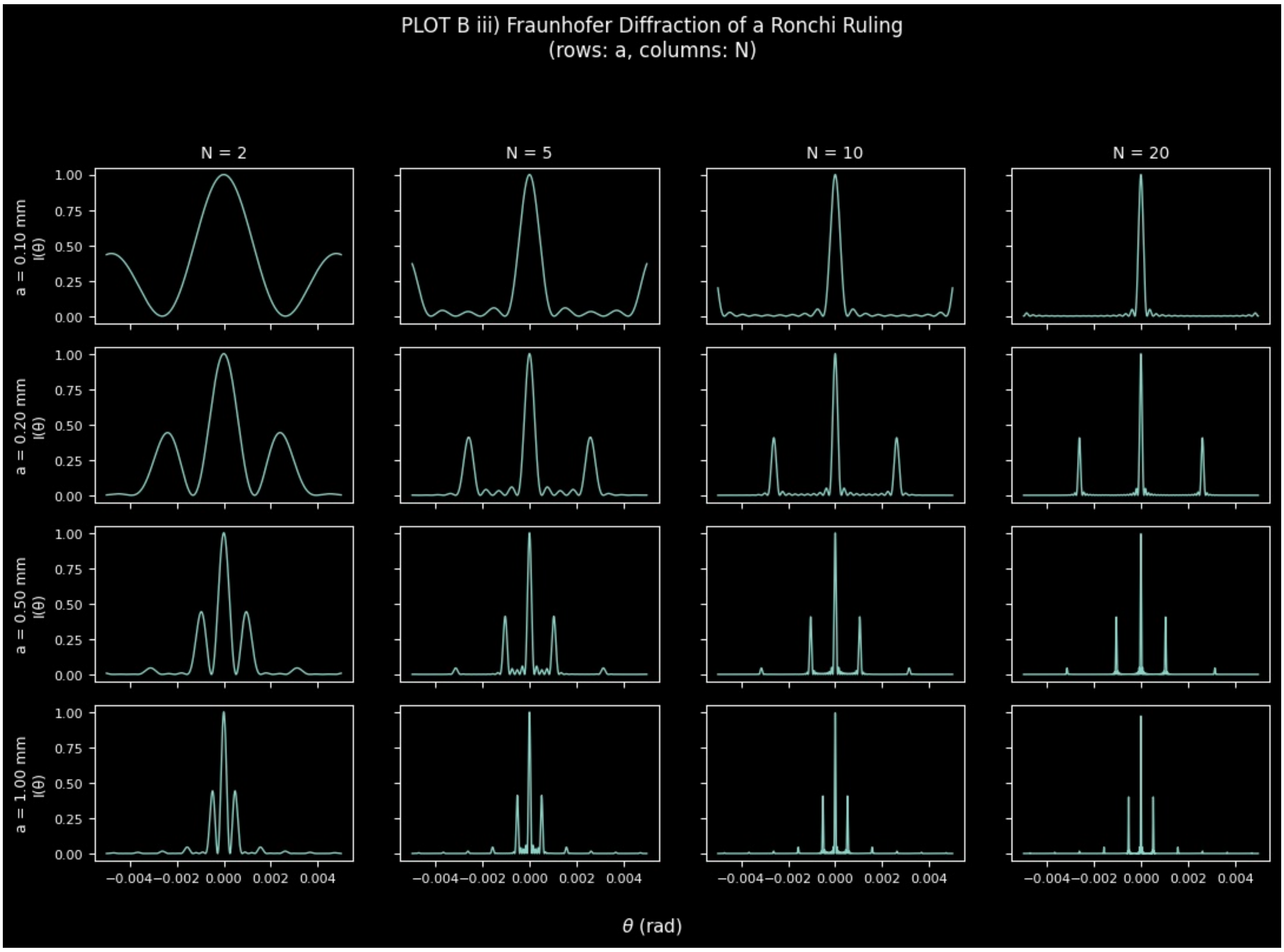
```
<Figure size 1600x800 with 0 Axes>
```



iii) Grid

In [13]:
```
# Angle range (rad) — chosen to show several diffraction orders
theta = np.linspace(-5e-3, 5e-3, 2000)    # ~±0.3 degrees

# Columns: different N
N_list = [2, 5, 10, 20]
```

```python
# Rows: different grating periods a (in mm)
a_list_mm = [0.1, 0.2, 0.5, 1.0]  # adjust to taste

# ----------------------------------------
# Create grid of plots: rows = a, cols = N
# ----------------------------------------

n_rows = len(a_list_mm)
n_cols = len(N_list)

# A4 in landscape: about 11.69 x 8.27 inches
fig, axes = plt.subplots(
    n_rows,
    n_cols,
    figsize=(11.69, 8.27),
    sharex=True,
    sharey=True
)

# If there's only one row/col, make axes always 2D-indexable
axes = np.atleast_2d(axes)

for i, a_mm in enumerate(a_list_mm):
    a_m = a_mm * 1e-3  # convert mm -> m
    for j, N in enumerate(N_list):

        ax = axes[i, j]
        I_vals = I(theta, N=N, a=a_m)

        ax.plot(theta, I_vals, lw=1.0)

        # Top row: label with N
        if i == 0:
            ax.set_title(f"N = {N}", fontsize=10)

        # First column: label with a
        if j == 0:
            ax.set_ylabel(f"a = {a_mm:.2f} mm\nI(θ)", fontsize=9)

        # Keep ticks small so everything fits nicely
        ax.tick_params(axis='both', which='both', labelsize=8)

# Common x-label
fig.text(0.5, 0.04, r"$\theta$ (rad)", ha='center', fontsize=11)

# Optional overall title
fig.suptitle("PLOT B iii) Fraunhofer Diffraction of a Ronchi Ruling\n(rows: a, columns: N)", fontsize=12)

# Make layout tight for A4 printing
fig.tight_layout(rect=[0.03, 0.07, 0.97, 0.93])

# Optional: save for printing
# plt.savefig("ronchi_grid_A4.pdf", dpi=300, bbox_inches="tight")

plt.show()
```



Fraunhofer Diffraction of a Ronchi Ruling: Effect of Slit Number N and Period a

Each panel shows the normalized far-field intensity $I(\theta)$ for a 1D Ronchi ruling, modeled as a grating of N identical slits with period a and duty cycle $b/a = 0.5$ (i.e., slit width $b = a/2$).

Rows correspond to different grating periods a (in mm), and columns correspond to different numbers of illuminated slits N. The diffraction pattern is computed from the standard N-slit Fraunhofer formula:

$$I(\theta) \propto \underbrace{\left[\frac{\sin(\beta)}{\beta}\right]^2}_{\text{single-slit envelope}} \times \underbrace{\left[\frac{\sin(N\alpha)}{N\sin(\alpha)}\right]^2}_{\text{N-slit interference}}$$

Variables Defined:

- $\alpha = \frac{1}{2}ka\sin\theta$
- $\beta = \frac{1}{2}kb\sin\theta$
- $k = 2\pi/\lambda$
- $\lambda = 525$ nm

Key Qualitative Features

- **Effect of increasing N (across columns):** The principal maxima become narrower and sharper, and the side lobes (subsidiary maxima) become more numerous but relatively weaker. This matches the behavior of the interference term: $[\sin(N\alpha)/(N\sin\alpha)]^2$.
- **Effect of increasing a (down rows):** The angular spacing between diffraction orders decreases approximately as $\Delta\theta \sim \lambda/a$. Larger periods produce more closely spaced peaks.
- **Single-slit envelope:** The overall envelope is set by $[\sin(\beta)/\beta]^2$. This broadens when b is smaller and narrows when b is larger. For a fixed duty cycle $b = a/2$, changing a simultaneously rescales both the order spacing and the envelope width.

Assumptions and Simplifications

Category	Description
Regime	**Fraunhofer (far-field)**: The pattern is the Fourier transform of the aperture; no Fresnel-region effects.

Geometry	**Scalar & 1D**: Polarization, finite slit height, and 2D structures are ignored.
Light Source	**Perfectly Coherent**: Single wavelength $\lambda = 525$ nm with no spectral bandwidth.
Optics	**Idealized**: No lens aberrations, misalignment, detector noise, or finite pixel size.
Normalization	**Per-subplot**: Intensities are normalized to their own maximum for shape comparison.

> **Note:** Under these idealized assumptions, the plots isolate the pure dependence of the Fraunhofer diffraction pattern on the grating period a and the number of illuminated slits N.

c)

Effect of varying the duty cycle b/a. How it affects the relative intensity of diffraction orders (including which orders may be suppressed).

I will be Fixing the N at 5 and a=0.20mm

and Vary the duty, b/a

In [14]:

```python
import numpy as np
import matplotlib.pyplot as plt
from fractions import Fraction as frac


def I_var_ba(theta, N, a, b_by_a, lam_nm=525, I0=1.0):
    """
    Calculates Fraunhofer intensity using the np.sinc identity.
    np.sinc(x) is defined as sin(pi*x)/(pi*x).
    """
    lam = lam_nm * 1e-9
    # s is the 'order' coordinate: (a*sin(theta))/lambda
    # When s is an integer, we are at an interference maximum.
    s = (a * np.sin(theta)) / lam

    # 1. Single Slit Envelope
    # Nulls occur when (b*sin(theta))/lambda is an integer.
    # This is equivalent to s * (b/a) being an integer.
    envelope = np.sinc(s * b_by_a)**2

    # 2. Interference Pattern
    # This identity avoids division by zero at sin(theta) = 0.
    # It normalizes the peak height to 1.0.
    interference = (np.sinc(N * s) / np.sinc(s))**2

    return I0 * envelope * interference

# --- Parameters ---
lam_nm = 525
N      = 10
a_mm   = 0.20
a      = a_mm * 1e-3

# Define duty cycles (b/a). b must be <= a.
b_by_a_list = [1.0, 1/2, 1/3, 1/4, 1/5,1/7]
theta = np.linspace(-np.pi/120, np.pi/120, 5000)

fig, axs = plt.subplots(len(b_by_a_list), 1, figsize=(8, 12),sharex=True)

for i, b_by_a in enumerate(b_by_a_list):
    I_vals = I_var_ba(theta, N, a, b_by_a, lam_nm)

    # Convert theta to 's' for the plot to see integer orders clearly
    s_vals = (a * np.sin(theta)) / (lam_nm * 1e-9)

    axs[i].plot(s_vals, I_vals, color='skyblue')
    axs[i].set_ylabel("Intensity")
    axs[i].legend(loc='upper right')
    axs[i].grid(alpha=0.3)
    axs[i].set_title(f"PLOT C) b/a={b_by_a:.1f} ")

    # Highlight where orders SHOULD be (integer s)
    # If a peak is missing at an integer s, you've found a missing order!



axs[-1].set_xlabel(r"Order $m$ ")
plt.xlim(-8, 8)
plt.tight_layout()
plt.show()
```

```
C:\Users\ahila\AppData\Local\Temp\ipykernel_19348\677395210.py:48: UserWarning: No artists with labels found to put in legend.  Note that artists whose label start with an underscore are ignored when legend() is called with no argument.
  axs[i].legend(loc='upper right')
```