```python
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import root_scalar
from scipy.optimize import fsolve


def compute_eigenpairs(L, x,etas, N=100):
    """Compute eigenvalues and eigenfunctions with corrected formulas."""
    # even_etas, odd_etas = compute_eta_values(L, N)
    odd_etas = etas[0::2]
    even_etas = etas[1::3]
    lambdas = np.zeros(N)
    phis = np.zeros((N, len(x)))

    for n in range(1, N+1):
        if n % 2 == 1:  # Odd n: use ODD eigenfunction (with even O· equation)
            idx = n // 2
            if idx < len(odd_etas):
                eta = odd_etas[idx]
                # Odd eigenvalue and eigenfunction (corrected: use sin for odd)
                lambdas[n-1] = 1 / (1 + L**2 * eta**2)
                denom = np.sqrt(1 + np.sin(2*eta)/(2*eta))
                phis[n-1, :] = np.sin(eta * x) / denom
        else:  # Even n: use EVEN eigenfunction (with odd O· equation)
            idx = n // 2 - 1
            if idx < len(even_etas):
                eta = even_etas[idx]
                # Even eigenvalue and eigenfunction (corrected: use cos for even)
                lambdas[n-1] = 1 / (1 + L**2 * eta**2)
                denom = np.sqrt(1 - np.sin(2*eta)/(2*eta))
                phis[n-1, :] = np.cos(eta * x) / denom

    return lambdas, phis
```