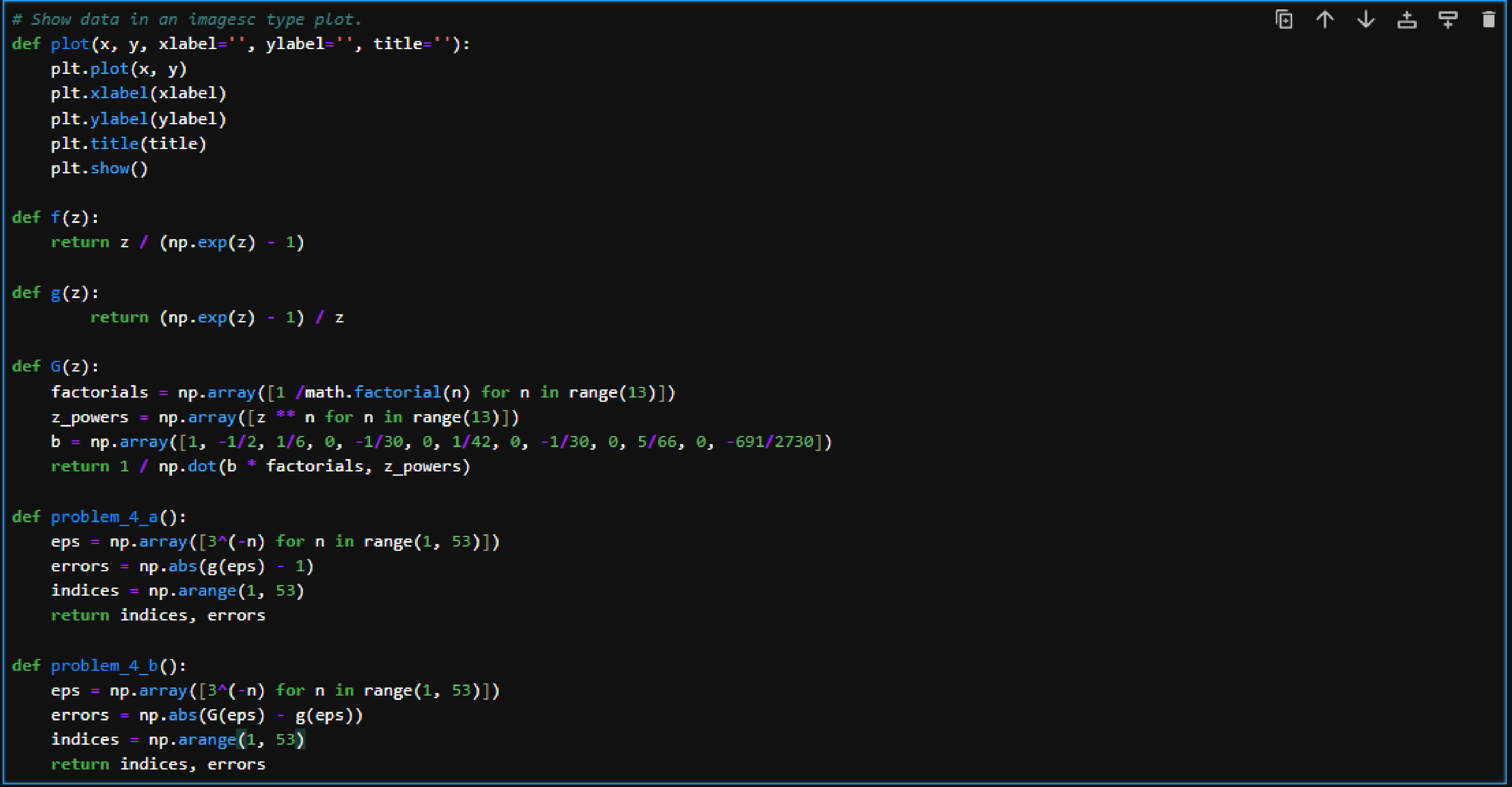
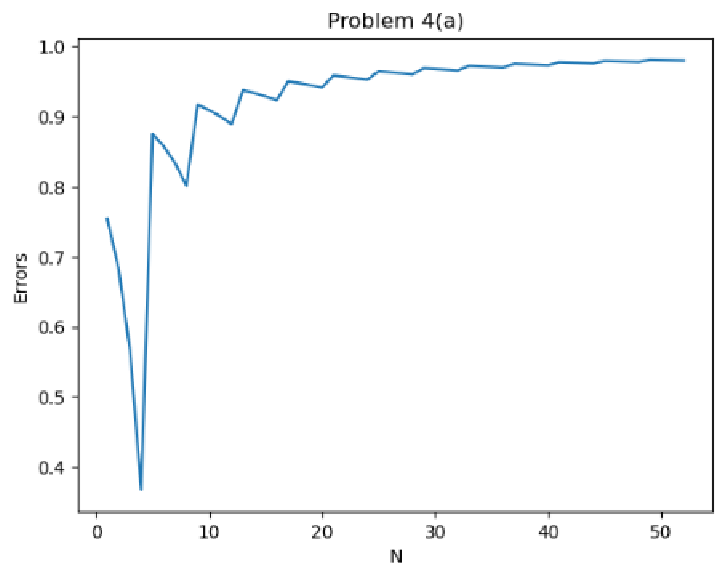
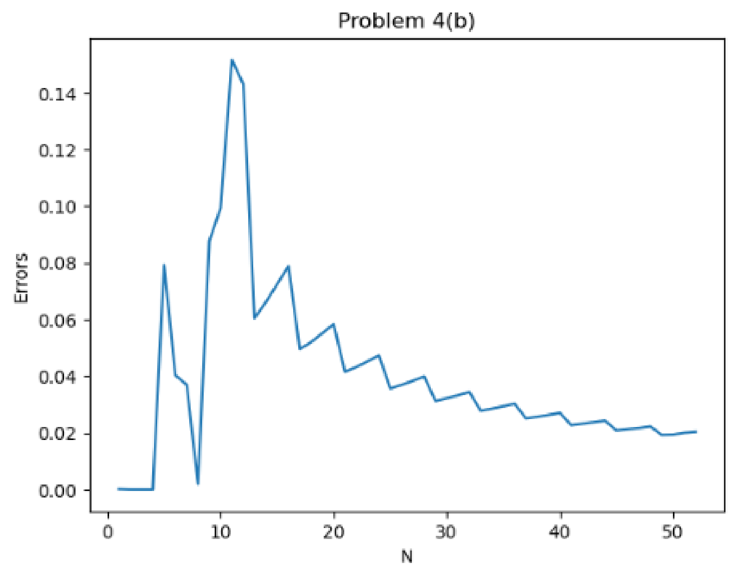


```
# Show data in an imagesc type plot.
def plot(x, y, xlabel='', ylabel='', title=''):
    plt.plot(x, y)
    plt.xlabel(xlabel)
    plt.ylabel(ylabel)
    plt.title(title)
    plt.show()

def f(z):
    return z / (np.exp(z) - 1)

def g(z):
        return (np.exp(z) - 1) / z

def G(z):
    factorials = np.array([1 /math.factorial(n) for n in range(13)])
    z_powers = np.array([z ** n for n in range(13)])
    b = np.array([1, -1/2, 1/6, 0, -1/30, 0, 1/42, 0, -1/30, 0, 5/66, 0, -691/2730])
    return 1 / np.dot(b * factorials, z_powers)

def problem_4_a():
    eps = np.array([3^(-n) for n in range(1, 53)])
    errors = np.abs(g(eps) - 1)
    indices = np.arange(1, 53)
    return indices, errors

def problem_4_b():
    eps = np.array([3^(-n) for n in range(1, 53)])
    errors = np.abs(G(eps) - g(eps))
    indices = np.arange(1, 53)
    return indices, errors
```

```
indices, errors = problem_4_a()
plot(x=indices, y=errors, xlabel='N', ylabel='Errors', title='Problem 4(a)')
```



```
indices, errors = problem_4_b()
plot(x=indices, y=errors, xlabel='N', ylabel='Errors', title='Problem 4(b)')
```


Problem 4(b)
Errors
N