In [ ]:
```python
import numpy as np
from scipy.optimize import minimize
from scipy.stats import ortho_group
import scipy as sp
import sympy as sm
import scipy.integrate as integrate
import matplotlib.pyplot as plt
import types
import math
import random
```

In [ ]:
```python
def mapping(x):
    #[1,5] -> [-1, 1]
    return 0.5 * x - 3/2
def unmapping(x):
    #[-1, 1] -> [1, 5]
    return 2*x+3
```

In [ ]:
```python
def coeff_eval(x: sm.core.symbol.Symbol, f : sm.Function, Cheby_poly : sm.Function, k: int):
    """
    Calcuates the Coefficients to the
    Chebyshev expansion.
    """
    inte = 0
    if k == 0:
        inte = (1./sm.pi) * f / (sm.sqrt(1 - x**2))
    else :
        inte = (2. / sm.pi) * f * Cheby_poly / (sm.sqrt(1 - x**2))

    return integrate.quad(sm.lambdify(x, inte, modules = ['numpy']), -1, 1)[0]


def Cheb (x : sm.core.symbol.Symbol, n_eval: int):
    """
    Returns the n-th Chebyshev polynomial.
    """
    j = sm.symbols('j', integer = True)
    n = sm.Symbol('n', integer = True)
    series = sm.Sum(sm.binomial(n, 2*j)* (x**2 - 1)**j * x**(n - 2*j), (j, 0, sm.floor(n/2)))
    return series.subs({n: n_eval}).doit()

def Cheb_expansion(f : sm.Function, k : int, cheb_dict: dict):
    x = sm.Symbol('x')
    cheb_coeff = np.zeros(k+1)
    cheb_funcs = []
    for i in range(k+1):
        cheb = Cheb(x, i)
        if i not in cheb_dict.keys():
            cheb_coeff[i] = coeff_eval(x, f, cheb, i)
            cheb_dict[i] = cheb_coeff[i]
        else :
            cheb_coeff[i] = cheb_dict[i]

        cheb_funcs.append(cheb)

    return cheb_coeff, cheb_funcs

def sup_norm(f_true : np.ndarray, f_approx : np.ndarray):
    f_diff = np.abs(f_true - f_approx)
    return np.max(f_diff)



### COMPUTING THE COEFFICIENTS
cheb_dict = {}

xs = np.linspace(1, 5, 100)
x = sm.Symbol('x')
f_map = 1/(2*x+3)

ks = range(1, 21, 1)
approxs = np.zeros((len(ks), 100))

lam_x = sm.lambdify(x, 1/x, modules=['numpy'])

for i, k in enumerate(ks):
    cheb_coeff, cheb_func = Cheb_expansion(f_map, k, cheb_dict)
    sum = 0
    lam_approx = 0
    for j in range(len(cheb_coeff)):
        sum += cheb_coeff[j] * cheb_func[j]
    lam_approx = sm.lambdify(x, sum, modules=['numpy'])
    approxs[i, :] = lam_approx(mapping(xs))


fig, axs = plt.subplots(1, 1, figsize = (16, 6))

for i in range(len(approxs)+1):
```

```python
    if i in [5, 10, 20]:
        axs.plot(xs, approxs[i-1, :], label = f'{ks[i-1]}')

axs.plot(xs, lam_x(xs), label = "True")

# Plotting Aesthetics
axs.set_title(f'Chebyshev Expansion Up to Degree {ks[-1]}')
axs.legend()
axs.grid(True)
axs.set_facecolor((0.9, 0.9, 0.9))
axs.set_xlabel("x")

plt.show()
```