```
# -*- coding: utf-8 -*-
"""
Created on Mon Apr 22 13:39:44 2024

@author: 2021n
"""

# get lgwts routine and numpy
import numpy as np;
import numpy.linalg as la;
import matplotlib.pyplot as plt

from scipy import integrate


def eval_composite_trap(M,a,b,f):
  x = np.linspace(a,b,M);
  h = (b-a)/(M-1);
  w = h*np.ones(M);
  w[0]=0.5*w[0]; w[M-1]=0.5*w[M-1];

  I_hat = np.sum(f(x)*w);
  return I_hat,x,w;

def eval_composite_simpsons(M,a,b,f):
  x = np.linspace(a,b,M);
  h = (b-a)/(M-1);
  # Explain why this defines the weights for Simpsons
  w = (h/3)*np.ones(M);
  w[1:M:2]=4*w[1:M:2];
  w[2:M-1:2]=2*w[2:M-1:2];

  I_hat = np.sum(f(x)*w);
  return I_hat,x,w;

f = lambda x: 1/(1+x**2)
fpp = lambda x: 2*(3*(x**2) - 1)/((x**2)+1)**3
fpppp = lambda x: 24*(5*(x**4) - 10*(x**2) + 1)/((x**2)+1)**5
print(fpp(0))


tol = 1e-4
a = -5
b = 5

Ms = np.arange(3,200,2); nM = len(Ms)
I_trap = np.zeros((nM,))
I_simp = np.zeros((nM,))
# storage for error
err_trap = np.zeros((nM,))
err_simp = np.zeros((nM,))
for iM in range(nM):
    M = Ms[iM]
    h = (b-a)/(M-1)
    I_trap[iM],_,_ = eval_composite_trap(M,a,b,f)
    I_simp[iM],_,_ = eval_composite_simpsons(M, a, b, f)
    err_trap[iM] = ((b-a)/12)*(h**2)*np.abs(fpp(0))
    err_simp[iM] = ((b-a)/180)*(h**4)*np.abs(fpppp(0))

    if err_simp[iM] < tol:
        print(err_trap[iM])

fig,ax = plt.subplots(1)
ax.semilogy(Ms,err_trap,'r--')
ax.set_xlabel('$M$')
ax.set_title('Trapezoid')
ax.set_ylabel('Relative Error');
plt.show()

fig,ax = plt.subplots(1)
ax.semilogy(Ms,err_simp,'b--')
ax.set_xlabel('$M$')
ax.set_title('Simpsons')
ax.set_ylabel('Relative Error');
plt.show()



I_quad,abserr,infodict = integrate.quad(f,-5,5, full_output=1, epsabs=1e-6)
print(infodict['neval'])
```