```
"""
Registration: xxxx;
Description: Orthonormality and recursion relation for Legendre functions
Author: AKB
"""

import numpy as np
from scipy.special import hermite, factorial, eval_hermite
from scipy.misc import derivative
import matplotlib.pyplot as plt
import scipy.integrate as sci
import warnings
warnings.filterwarnings("ignore")

# Feed the value of n, m, lower and upper limit, #of points (from keyboard)
n = 10; m = n+1; low = -1; up = -low; Np = 100;
x = np.linspace(low, up, Np);

""" Create Poly1D Legendre polynomial and derivatives
Note that other way of defining is using eval_hermite as  hn = eval_hermite(n,x); hm =
eval_hermite(m,x);
hnm1 = eval_hermite(n-1,x); hnp1 = eval_hermite(n+1,x); then define lhs = hnp1, rhs =
2*x*hn-2*n*hnm1
yields identical result
"""
hn = hermite(n); hm = hermite(m);
hnm1 = hermite(n-1); hnp1 = hermite(n+1);
hnprime = derivative(hn, x, 1e-6)  # spacing=10^-6

# Logical case switch for different recursion relations to choice from
ortho=1; recl1=1; recl2=1; recl3=1; recl4=1;
print('Compare maximum of |lhs-rhs| (L1 norm) to zero for n = ', n)

if(ortho): # \int Pn(x)Pm(x) = 2/(2n+1)*delta(nm)
  def hnhm(x,n): return hn(x)*hm(x)*np.exp(-x**2)/(np.sqrt(np.pi)*(2**n)*factorial(n))
  I = sci.quad(hnhm, -np.inf, np.inf, args=(n,))[0]
  print('Orthonormality : \int_(-inf)^(inf) H_',n,'(x)H_',m,'(x)*e^(-x^2)dx = ', I)

if(recl1): #H(n+1)(x) = 2xH(n)(x) - 2nH(n-1)(x)
  lhs = hnp1(x)
  rhs = 2*x*hn(x)-2*n*hnm1(x)
  plt.plot(x, lhs, 'o', label='LHS'); plt.plot(x, rhs, label='RHS'); plt.legend();
plt.show()
  print('Maximum of H(n+1)(x) - 2xH(n)(x) + 2nH(n-1)(x) = ', abs(max(lhs-rhs)))

if(recl2): #H(n)(-x) = (-1)^n Hn(x)
  lhs = hn(-x)
  rhs = (-1)**n*hn(x)
  plt.plot(x, lhs, 'o', label='LHS'); plt.plot(x, rhs, label='RHS'); plt.legend();
plt.show()
  print('Maximum of H(n)(-x) - (-1)^n Hn(x) = ', abs(max(lhs-rhs)))

if(recl3): # Hn'(x) = 2nH(n-1)(x)
  lhs = hnprime
  rhs = 2*n*hnm1(x)
  plt.plot(x, lhs, 'o', label='LHS'); plt.plot(x, rhs, label='RHS'); plt.legend();
plt.show()
  print('Maximum of dHn(x)/dx - 2nH(n-1)(x) = ', abs(max(lhs-rhs)))

if(recl4): #Turan's Inequality Hn(x)^2-H(n-1)(x)H(n+1)(x) = (n-1)! Sum_{i=0}^{n-1} (2^{n-
i})/i! * Hi(x)^2
  lhs = hn(x)**2 - hnm1(x)*hnp1(x)
  rhs = factorial(n-1)*sum(2**(n-i)/factorial(i)*hermite(i)(x)**2 for i in range(0, n))
  plt.plot(x, lhs, 'o', label='LHS'); plt.plot(x, rhs, label='RHS'); plt.legend();
plt.show()
  print('Turans Inequality Satisfied till ', abs(max(lhs-rhs)))
```

```
"""
Results (2 Sets) :
Compare maximum of |lhs-rhs| (L1 norm) to zero for n =  2
Orthonormality : \int_(-inf)^(inf) H_ 2 (x)H_ 2 (x)*e^(-x^2)dx =  0.9999999999999998
Orthonormality : \int_(-inf)^(inf) H_ 2 (x)H_ 3 (x)*e^(-x^2)dx =  0.0
Maximum of H(n+1)(x) - 2xH(n)(x) + 2nH(n-1)(x) =  2.1316282072803006e-14
Maximum of H(n)(-x) - (-1)^n Hn(x) =  0.0
Maximum of dHn(x)/dx - 2nH(n-1)(x) =  2.1188952814554796e-09
Turans Inequality Satisfied till  2.842170943040401e-14

Compare maximum of |lhs-rhs| (L1 norm) to zero for n =  39
Orthonormality : \int_(-inf)^(inf) H_ 39 (x)H_ 39 (x)*e^(-x^2)dx =  1.0000000000000064
Orthonormality : \int_(-inf)^(inf) H_ 39 (x)H_ 40 (x)*e^(-x^2)dx =  0.0
Compare maximum of |lhs-rhs| (L1 norm) to zero for n =  10
Maximum of H(n+1)(x) - 2xH(n)(x) + 2nH(n-1)(x) =  5.820766091346741e-11
Maximum of H(n)(-x) - (-1)^n Hn(x) =  0.0
Maximum of dHn(x)/dx - 2nH(n-1)(x) =  1.0957737686112523e-05
Turans Inequality Satisfied till  9.5367431640625e-07
"""
```