```python
"""
Created by Ye Cheng (Lewis) Chen and Jesko Sirker
The non-interacting disordered SSH model
Calculation for Loschmidt echo using SPH and formula |det(1-C+C*exp(-iHt))|^2
"""
import numpy as np
import argparse

# parameters for entanglement calculations
parser = argparse.ArgumentParser(description=
'Calculation of Loschmidt echo for disordered fermions')
parser.add_argument('-L', type=int, default=4,
                    help='# system length L')
parser.add_argument('-W', type=float, default=0.0,
                    help='width box potential disorder')
parser.add_argument('-delta', type=float, default=-0.3,
                    help='# quench parameter delta')
parser.add_argument('-dt', type=float, default=0.01,
                    help='# discrete time interval first part')
parser.add_argument('-tint', type=float, default=0,
                    help='# maximum time range first part')
parser.add_argument('-tmax', type=float, default=20,
                    help='# maximum time range second part')
parser.add_argument('-sample', type=int, default=1,
                    help='# samples')
parser.add_argument('-openbc', type=int, default=1,
                    help='OBC = 1, PBC = 0')
args=parser.parse_args()

# construct Hamiltonian for SSH model with disorder in diagonal elements
# diagnonal elements are random numbers in [-W,W]
def construct_APDW(L,W):
    if args.W != 0.0:
        a = 2*W * np.random.random_sample(L) - W
    else:
        a = np.zeros(L)
    A = np.diag(a,0)
    return A

# construct single-particle Hamiltonian for SSH model
def construct_SPH(delta,L,openbc):
    H = np.zeros((L,L))
    for i in range(0,L-1):
        H[i,i+1]=1.0-delta*(-1)**i
        H[i+1,i]=1.0-delta*(-1)**i
    if openbc == 0:
        H[0][-1]=1.0+delta*(-1)**L
        H[-1][0]=1.0+delta*(-1)**L
    return H

# construct unitary time evolution operator Uexp(-iDt)U*
def construct_U(v,U,t):
    Ut = np.dot(U.conj(),np.dot(np.diag(np.exp(-1j*v*t)),(U.transpose()).conj()))
    return Ut
```