```python
from TimeIntegration import getbc, get_eigenval
from FixedVariables import rho, v, p, nw, xmax, xmin
from ComputeTimeStep import computeTimeStep
from InputVariables import ngc
import numpy as np

def flux(h):
    F = [0, 0, 0]
    F[0] = h[1]
    F[1] = rho(h)*np.square(v(h)[0]) + p(h)
    F[2] = (h[-1] + p(h))*v(h)[0]
    return F

def IntegrateTimeLF(x,nx):
    dx = (xmax - xmin) / (nx - 1)
    w = getbc(x)
    lam = get_eigenval(w, nx)
    dt = computeTimeStep(lam, dx, nx)
    w_adv = [[0 for l in range(nw)] for m in range(nx+ngc*2)]
    for k in range(nw):
        w_adv[0][k] = w[0][k]
        w_adv[-1][k] = w[-1][k]
    for i in range(ngc, nx + ngc):
        for j in range(nw):
            w_adv[i][j] = (w[i+1][j] + w[i-1][j])/2 - (dt/(2*dx))*(flux(w[i+1])[j] - flux(w[i-1])[j])
    print(w_adv)
    return dt, w_adv
```