```
import numpy as np

np.set_printoptions(precision=3)

ABD = np.array([
    [  5.037e+07,  1.809e+06,  0        ,  3.231e+03,  0        ,  0        ],
    [  1.809e+06,  5.037e+07,  0        ,  0        , -3.231e+03,  0        ],
    [  0        ,  0        ,  2.640e+06,  0        ,  0        ,  0        ],
    [  3.231e+03,  0        ,  0        ,  1.511    ,  5.400e-02,  0        ],
    [  0        , -3.231e+03,  0        ,  5.400e-02,  1.511    ,  0        ],
    [  0        ,  0        ,  0        ,  0        ,  0        ,  7.900e-02]])

a = 1.5 #m
b = 1.5 #m

q0 = 1 #N/m/m

def Qmn(m, n): #uniform loading
    return (16*q0)/(np.pi**2 * m * n)

def mn(m_range, n_range):
    m, n = np.meshgrid( np.arange(m_range)+1, np.arange(n_range)+1 )
    return m, n

def a_mn(m, n, Nx=0, Ny=0): #Navier SS1 case
    A = (m*np.pi)/a
    B = (n*np.pi)/b

    c11 = ABD[0,0]*(A**2) + ABD[2,2]*(B**2)
    c12 = (ABD[0,1] + ABD[2,2])*A*B
    c13 = -ABD[0,3]*(A**3) - (ABD[0,4] + 2*ABD[2,5])*A*(B**2)
    c22 = ABD[2,2]*(A**2) + ABD[1,1]*B**2
    c23 = -ABD[1,4]*(B**3) - (ABD[0,4] + 2*ABD[2,5])*(A**2)*B
    c33 = ABD[3,3]*(A**4) + 2*(ABD[3,4] + 2*ABD[5,5])*(A**2)*(B**2) +
ABD[4,4]*(B**4)

    s33 = Nx*A**2 + Ny*B**2

    a0 = c11*c22 - c12*c12
    a1 = c12*c23 - c13*c22
    a2 = c13*c12 - c11*c23

    amn = c33 + (c13*a1 + c23*a2)*a0**-1

    return amn

def Wmn(m, n):
    return Qmn(m, n) * a_mn(m, n)**-1
```

```
def w_o(x, y, precision=3):
    W = 0
    W_new = 1

    size = 1
    while size < 30:
        p = np.floor( -np.log10(np.abs(W_new-W)) )
        if p == np.inf: p = precision

        print('size: %d \tprecision: %d\tw: %.10f' %(size, p, W_new))

        W = W_new

        m, n = mn(size, size)

        W_new = np.sum( Wmn(m, n) * np.sin(m*np.pi*x*a**-1) *
np.sin(n*np.pi*y*b**-1) )

        size += 1

np.seterr(divide='ignore')
w_o(a/2, b/2, precision=5)
```