

# Prediseño de diques de abrigo

Prof. María Clavero Gilabert

Dept. Mecánica de Estructuras e Ingeniería Hidráulica

E.T.S.I. Caminos, Canales y Puertos

Universidad de Granada

Se importan los paquetes necesarios para ejecutar el script:

- numpy:
- pyplot:
- math:
- fsolve:

```
[2] import numpy as np
import matplotlib.pyplot as plt
from array import array
import math
from scipy.optimize import fsolve
```

La función para resolver la ecuación de la dispersión se puede incluir dentro del script general mediante la orden "def".

```
[3] #=====
# Script for wave number calculation

def wavnum(t, h):
    g = 9.81
    gamma = (2 * np.pi / t) ** 2 * h / g
    sigma2 = (2 * np.pi / t) ** 2

    def func(k):
        return sigma2 / g - k * np.tanh(k * h)

    if gamma > np.pi ** 2.:
        ko = np.sqrt(gamma / h)
        k = fsolve(func, ko)
```

```

else:

    ko = gamma / h
    k = fsolve(func, ko)

k = abs(k)

return k

#=====

```

El primer paso es conocer los datos iniciales de proyecto, algunos provienen del estudio de clima marítimo y el emplazamiento y otros se seleccionarán como criterio de diseño:

[22]

```

# Initial data

g=9.81
gammaw=1.025 # Ton/m^3
gammaw=2.35 # Concrete specific weight

#Hs0=9.50
#T=19 # Mean wave period

theta0=0*np.pi/180 # Incidence angle

h=10 # Water depth in front of the breakwater
Bb=8 # Caisson foundation width
S1=1.5 # Caisson foundation thickness
S2=3 # Toe berm thickness

d=h-S1-S2 # Water depth over the toe berm
hp=h-S1 # Depth of submerged caisson
tanfondo=1/60 # Bottom slope angle
Fc=3.5 #Freeboard level
Fm=1.5 # Dock level (leeside)
botaolas=3 # Crown wall width (B/botaolas)

```

Vamos a considerar varias parejas de valores H-T para el prediseño. Se leen de un archivo de texto y se visualizan en un gráfico:

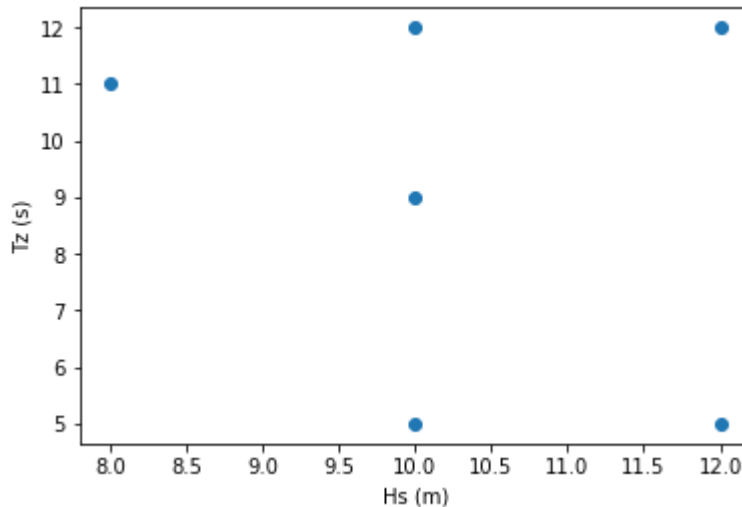
[23]

```

oleaje=np.loadtxt('oleaje_disenio.txt',comments='#')
Hs0=oleaje[:,0]
T=oleaje[:,1]
plt.plot(Hs0,T,'o')
plt.xlabel('Hs (m)')
plt.ylabel('Tz (s)')

```

Text(0,0.5,'Tz (s)')



Si los datos del clima marítimo están tomados de aguas profundas hay que propagar el oleaje:

```
[24] # Wave propagation
      caso=0 ## se selecciona el valor de H-T para el que diseñar

      sigma=2*np.pi/T[caso] # Angular frequency
      L0=9.81*(T[caso]**2)/(2*np.pi); #
      k0=(2*np.pi)/L0 # Deep water wave number
      k1=wavnum(T[caso],h) # Wave number from dispersion equation at depth h
      L1=2*np.pi/k1 # Wave length at depth h
      theta1=math.asin(k0*math.sin(theta0)/k1) # Snell Law
      Kr=math.sqrt(math.cos(theta0)/math.cos(theta1)) # Refraction Coefficient

      cg0=0.5*L0/T[caso] # Deep water group celerity
      cg1=0.5*sigma/k1*(1+2*k1*h/(math.sinh(2*k1*h))) # Group velocity at depth
      Ks=math.sqrt(cg0/cg1) # Shoaling coefficient

      Hsp=Hs0[caso]*Ks*Kr # Total wave height in front of the breakwater

      print('La altura de ola significativa es')
      print(Hs0[caso])
      print('La altura de ola a pie de dique es')
      print(Hsp)
```

La altura de ola significativa es

8.0

La altura de ola a pie de dique es

8.1053745297489

Conocidos los valores de clima marítimo de diseño, se debe seleccionar el método de cálculo a aplicar (Goda o Takahashi):

```

[25] # Method selection

from VB_methods import Goda
from VB_methods import Takahashi

R=1 # reflection coefficient

Hmax=1.8*Hsp

H_max_L=Hmax/L1
Hrot=(0.11+0.03*(1-R)/(1+R))*math.tanh(k1*h) # if Hmax/L<Hrot no depth-br

if H_max_L>Hrot: # check depth-breaking condition --> if positive, Takaha

    print('El oleaje rompe por fondo y se aplica Takahashi')
    Hc=0.11*math.tanh(k1*h)*L1
    [p1,p2,p3,p4,pu] = Takahashi(theta1,Hc,h,L1,Hsp,tanfondo,d,hp,Fc,Bb,g

else:
    Coef_a=(h-d)/h
    Coef_b=Bb/L1

    if Coef_a >= 0.3 and Coef_b >= 0.01:

        print('El oleaje rompe sobre la berma y se aplica Takahashi')
        Hc=0.11*math.tanh(k1*h)*L1
        [p1,p2,p3,p4,pu] = Takahashi(theta1,Hc,h,L1,Hsp,tanfondo,d,hp,Fc,

    else:

        print('El oleaje no ha roto por fondo ni por berma')
        print('Es necesario comprobar que se está en el rango de aplicaci
        Coef_c=h/L1
        Coef_d=Hsp/L1

        print('La profundidad relativa h/L es')
        print(Coef_c)
        print('El peralte H/L es')
        print(Coef_d)
        aux_method=int(input('Confirmación de aplicación de Goda (si=1/no
        if aux_method==1:
            Hc=Hmax
            [p1,p3,p4,pu] = Goda(theta1,Hc,h,Hsp,tanfondo,L1,d,hp,gammaw,
        else:
            print('No se puede aplicar el Método de Goda')
            Hc=0.11*math.tanh(k1*h)*L1
            [p1,p2,p3,p4,pu] = Takahashi(theta1,Hc,h,L1,Hsp,tanfondo,d,hp

```

El oleaje rompe por fondo y se aplica Takahashi

Por último, se calculan las fuerzas y momentos y se obtiene la anchura óptima de la base del cajón vertical:

```
[26] # Forces

Fh=0.5*(p1+p3)*hp+0.5*(p1+p4)*Fc
# Fv=0.5*pu*B
# W=(B/botaolas)*(Fc-Fm)*gammas+B*Fm*gammas+B*hp*(gammas-gammaw)

# Sliding

CS=1.2 # safety coefficient

B1_des=CS*Fh/0.6;
B2_des=(1/botaolas)*(Fc-Fm)*gammas+Fm*gammas+hp*(gammas-gammaw)-0.5*pu;
B_des=B1_des/B2_des;

# Moments
Mh=0.5*(p1-p3)*(2/3)*(hp**2)+0.5*p3*(hp**2)+0.5*(p1-p4)*Fc*(hp+Fc/3)+p4*Fc
Mh_comprobac=(1/6)*(2*p1+p3)*(hp**2)+0.5*(p1+p4)*hp*Fc+(1/6)*(p1+2*p4)*(Fc
# Mv=0.5*pu*B*(2/3)*B
# Mw=(B/botaolas)*(Fc-Fm)*gammas*(B-B/(2*botaolas))+B*Fm*gammas*B/2+B*hp*

## Overturning

B1_vuelco=Mh*CS
B2_vuelco=(1/botaolas)*(Fc-Fm)*gammas*(1-1/(2*botaolas))+Fm*gammas*1/2+hp
B_vuelco=math.sqrt(B1_vuelco/B2_vuelco)

# Optimum width
Bmax=max(B_des,B_vuelco)
```

También se puede calcular el tamaño de las piezas de la berma de pie:

```
[27] # Peso de las piezas de la berma de protecci?n del dique vertical
Aw=0.06819; # Cubos de hormig?n
Bw=-0.5148; # Cubos de hormig?n
tanb=1/1.5; # Pendiente de la berma
Ir=tanb/((math.sqrt(Hmax/L0)))
Ir0=2.654*tanb
phi=Aw*(Ir-Ir0)*math.exp(Bw*(Ir-Ir0))
phi_s=phi*((math.cosh(k1*(h-d))/math.cosh(k1*h))*2/(1+R))**6)
```

```
Sr=gammas/gammaw  
W=phi_s*gammaw*(Hc**3)*Sr/((Sr-1)**3)
```

Finalmente se obtienen en pantalla los valores principales del diseño:

```
[28] print('El valor de la anchura del cajón es')  
      print(Bmax)  
      print('El peso de las piezas es')  
      print(W)
```

```
El valor de la anchura del cajón es  
[11.64800123]  
El peso de las piezas es  
[3.44296094]
```