# NTF 4 - Portos

## Hidráulica Marítimica

Aluno: Francisco José Matos Nogueira Filho

Matricula: 384962

Entrée [1]:

```
import numpy as np
from numpy import pi
from IPython.display import Markdown as md
```

## Dados da onda em Alto Mar

Entrée [2]:

```
## Dados da onda em Alto Mar

H0 = 4 #m
T = 10 #s
g = 9.81 #m/s^2
```

### Redimencionamento da Altura de RAM

Entrée [3]:

```
def comprimentoDeOnda(Periodo):
    return (9.81 * Periodo**2)/(2 * pi)

L0 = comprimentoDeOnda(T)

#Resposta
md("$L_0 = %f$ M"%(L0))
```

Out[3]:

$L_0 = 156.130999$ M

Entrée [4]:

```
def Celeridade(Comprimento,T):
    return Comprimento/T

C0 = Celeridade(L0,T)

#Resposta
md("$C_0 = %f$ m/s"%(C0))
```

Out[4]:

$C_0 = 15.613100$ m/s

Entrée [5]:

```
_sigma = 2*pi/T
md("$\sigma = %f s^{-1}$"%(_sigma))
```

Out[5]:

$\sigma = 0.628319 s^{-1}$

Aproximaremos h pela equação de Hunt

**[1]** $y = \frac{\sigma^2 h}{g}$

**[2]**
$(Kh)^2 = y^2 + \frac{y}{1+0.666*y^2+0.355*y^3+0.161*y^4+0.063*y^4+0.022*y^5+0.006$

**[3]** $Kx = \frac{KH}{h}$

**[4]** $L_x = L_0 \tanh(K_x H_x)$

**[5]** $n = 0.5 \left[1 + \frac{Kd}{senh(2Kd)}\right]$

**[7]** $h = H_0 \sqrt{\frac{b_0}{b}} \sqrt{\frac{1}{2} \frac{1}{n} \frac{C_0}{C}}$

Usarei uma função `minimize` do pacote de calculo númerico [scipy (https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.minimize.html)](https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.minimize.html) para minimizar a fu

Entrée [6]:

```
from scipy.optimize import minimize, newton_krylov, broyden1, curve_fit
 #m

def calcularY(sigma,h,g): #1
    return (sigma**2 * h)/g
def calculateKH(y): #2
    return (y**2 +y/(1+ 0.666*y + 0.355*y**2 + 0.161*y**3 + 0.063*y**4 + 0.022*y**5
def calculateKx(KH,h): #3
    return KH/h
def calculateLx(L0, Kx, Hx): #4
    return L0*np.tanh(Kx*Hx)

def calculateN(K,d): #5
    return 0.5 * (1 + (2*K*d)/np.sinh(2*K*d))
def calculateC(Lx,T): #6
    return Lx/T
def calculateH(H0,n,C0,C): #7
    return H0 * (1 * 0.5* 1/n * C0/C)**0.5
```

Entrée [7]:

```
def fu(h):
    H_desejado = 5 #m
    y = calcularY(_sigma,h,g)
    KH = calculateKH(y)
    Kx = calculateKx(KH,h)
    Lx = calculateLx(L0, Kx, h)
    n = calculateN(Kx, h)
    C = calculateC(Lx,T)
    H = calculateH(H0,n,C0,C)

    return (H-H_desejado)**2
x0 = 1 #Palpite inicial
solv = minimize(fu,x0,method='SLSQP')
h = (solv.x)[0]
```

Entrée [8]:

```
y = calcularY(_sigma,h,g)
KH = calculateKH(y)
Kx = calculateKx(KH,h)
Lx = calculateLx(L0, Kx, h)
n = calculateN(Kx, h)
C = calculateC(Lx,T)
H = calculateH(H0,n,C0,C)
```

Entrée [9]:

```
md("""
$ H_x = %f $m \n
$y = %f $m \n
$ Kh = %f $ \n
$ K_x = %f $ \n
$ L_x = %f $m \n
$ n = %f $ \n
$ C = %f $ \n
$ H = %f $m \n
""" % (h,y,KH,Kx,Lx,n,C,H))
```

Out[9]:

$H_x = 2.856527$m

$y = 0.114955$m

$Kh = 0.345697$

$K_x = 0.121020$

$L_x = 51.921988$m

$n = 0.962279$

$C = 5.192199$

$H = 4.999923$m

## Energia de Onda

$$E = \overline{E}L = \frac{\rho g H^2 L}{8}$$

Entrée [10]:

```
rho = 1034
H_rec = 5
E = (1034 * g * H_rec**2 * Lx)/8

# Para comprimento de crista = 1m
md("""$E = %f $ Joules \n
    ou \n $E = %f $ tf m """%(E, E/(9.81*1000)))
```

Out[10]:

$E = 1645852.367678$ Joules

ou

$E = 167.772922$ tf m

## Força de impacto da onda no topo do RAM

$$u = \frac{agKcosh[K(d+z)]}{\sigma cosh(Kd)} cos(Kx - \sigma t)$$

Entrée [11]:

```
d = 12 - h
a = H_rec/2
u = (a * g * Kx * np.cosh(Kx * (d - h)))/(_sigma * np.cosh(Kx * d))

md("$ u_{ max\; em\; z = -2.85} = %f $ m/s" % u)
```

Out[11]:

$u_{max\; em\; z=-2.85} = 3.671537$ m/s

Entrée [12]:

```
#Para coeficiente hidrodinâmico C = 1.4
A = 1
F = 0.5 * 1.4 * rho * g * u**2 * A
md("$ F = %f $ N" % F)
```

Out[12]:

$F = 95715.730145$ N