02/18/18 01:12:05 /home/nsamba/projects/barrier.jl

```julia
#!/usr/bin/julia

type Barrier
    height
    base_width
    base_length
    base_thickness
    wall_thickness
end
barrier=Barrier(0,0,0,0,0)
type Constants
    spec_water
    spec_mat
    spec_base
    spec_soil
    mat_shear
    Cb
    Cf
    Kp
    Sb
end
constants=Constants(64,0,0,0,0,0,0,0,0)
base_len(x)=0.40*x
base_thick(x)=0.44*x
Area(force,shear)=force/shear
FArea(P,Sb)=P/Sb
BArea(a,e)=a*e
F_sta(pg,H)=4*0.5*pg*H^2
weight(spec_wt,area,height)=spec_wt*area*height
Fbuoy(spec_wt,area,height)=spec_wt*area*height
Fv(wall,basew,Wwat,wallw,fb)=basew+wall+Wwat+wallw-fb
Fr(Cf,fv)=Cf*fv
Fc(Cb,B)=Cb*B
Fp(kp,spec_soil,spec_water,t,len)=0.5*(kp*(spec_soil-spec_water)+spec_water)*(t^2)*len
function Fres(constants,barrier,Fst)
    wall_area=Area(Fst,constants.mat_shear)
    wall=weight(constants.spec_mat,wall_area,barrier.height)
    wc_area=(barrier.wall_thickness)^2
    wall_con=weight(constants.spec_soil,wc_area,barrier.height)
    base_area=BArea(barrier.base_width,barrier.base_length)
    basew=weight(constants.spec_base,base_area,barrier.base_thickness)
    pct=basew
    println("weight of base: $pct")
    val=(2/3)*(barrier.base_width-barrier.wall_thickness)
    val1=(1/3)*barrier.base_width
    water_base=BArea(val,barrier.base_length)
    Wwat=weight(constants.spec_water,water_base,barrier.height)
    fb=Fbuoy(constants.spec_water,base_area,barrier.base_thickness)
    println("Bouyant Force: $fb")
    fv=Fv(basew,wall,Wwat,wall_con,fb)
    println("Vertical force : $fv")
    friction=Fr(constants.Cf,fv)
    fc=Fc(constants.Cb,barrier.base_width)
    fp=Fp(constants.Kp,constants.spec_soil,constants.spec_water,barrier.base_thickness,barrier.base_length)
    println("passive soil pressure: $fp")
    resf=friction+fc+fp
    return [resf,fv]
end
function design(barrier,constants)
    #for redesign of barrier
    base_width=barrier.base_width
    println("Assuming base_length = 0.80 of base_width")
    barrier.base_length=base_len(barrier.base_width)
    println("Assuming base_thickness= 0.70 of base_width")
    barrier.base_thickness=base_thick(barrier.base_width)
    temp1=constants.spec_water
    temp2=barrier.height
    Fst=F_sta(temp1,temp2)
    println("hydrostatic force: $Fst")
    println("Calculating opposing forces")
    Fresist=Fres(constants,barrier,Fst)
    ans=Fresist[1]/Fst
    tempv=Fresist[1]
    println("Resisting force: $tempv")
    println(Fresist[1])
    return [ans,Fresist[2]]
end
function ch_base(barrier,coef)
    height=barrier.height
    coef=coef+0.05
    barrier.base_width=coef*height
    return coef
end
heel(x,z)=(2/3)*(x-z)
toe(x,z)=(1/3)*(x-z)
M_sta(fst,H,t)= fst*((H+t)/3)
M_fp(fpee,t)= fpee*(t/3)
M_bouy(fb1,fb2,B) = (fb1*(B/3)) + (fb2*(2/3)*B)
Mbse(bw,B)=bw*(B/2)
Mwallwt(Fwt,C,twall)= Fwt*(C+(twall/2))
Mwh(waterw,B,Ah)=waterw*(B-(Ah/2))
fboy(a,H,twall,ah,t,len)= a*((H)*(0.5*twall)+(ah*(twall/2)*(t)))*len
function Mres(barrier,constants)

```

```julia
    fp=Fp(constants.Kp,constants.spec_soil,constants.spec_water,barrier.base_thickness,barrier.base_length)
    base_area=BArea(barrier.base_width,barrier.base_length)
    basew = weight(constants.spec_base,base_area,barrier.base_thickness)
    Mbase = Mbse(basew,barrier.base_width)
    fst=F_sta(constants.spec_water,barrier.height)
    wall_area=Area(fst,constants.mat_shear)
    wall=weight(constants.spec_mat,wall_area,barrier.height)
    wc_area=(barrier.wall_thickness)^2
    wall_con= weight(constants.spec_soil,wc_area,barrier.height)
    wallwt = wall_con+wall
    Mwwt = Mwallwt(wallwt,toe(barrier.base_width,barrier.wall_thickness),barrier.wall_thickness)
    water_area = BArea(heel(barrier.base_width,barrier.wall_thickness),barrier.base_length)
    wh = weight(constants.spec_water,water_area,barrier.height)
    Mwheel=Mwh(wh,barrier.base_width,heel(barrier.base_width,barrier.wall_thickness))
    Mfp = M_fp(fp,barrier.base_thickness)
    Mresisting=Mfp+Mwwt+Mbase+wh
    println("And the resisting Moment is: $Mresisting")
    return Mresisting
end
function overturning(barrier,constants)
    #barrier.base_thickness = base_thick(barrier.base_width)
    #barrier.base_length = base_len(barrier.base_width)
    fst=F_sta(constants.spec_water,barrier.height)
    Mst=M_sta(fst,barrier.height,barrier.base_thickness)
    hel=heel(barrier.base_width,barrier.wall_thickness)
    toy=toe(barrier.base_width,barrier.wall_thickness)
    fbuoy1=fboy(constants.spec_water,barrier.height,barrier.wall_thickness,hel,barrier.base_thickness,barrier.base_length)
    fbuoy2=fboy(constants.spec_water,barrier.height,barrier.wall_thickness,toy,barrier.base_thickness,barrier.base_length)
    Mbouy=M_bouy(fbuoy1,fbuoy2,barrier.base_width)
    Mov=Mst+Mbouy
    println("And the overturning Moment is: $Mov")
    Mresist=Mres(barrier,constants)
    ans=Mresist/Mov
    qns=[ans,Mres,Mov]
    return qns
end
function main(barrier,constants)
    println("Design Flood Elevation in ft")
    height=readline(STDIN)
    barrier.height=parse(Int8,height)
    println("Assuming base_width = wall_height")
    barrier.base_width=0.5*barrier.height
    println("Wall thickness in ft")
    wall_t=readline(STDIN)
    barrier.wall_thickness=parse(Float32,wall_t)
    println("Establishing constants....")
    println("Insert soil specific weight in Ib/ft3 ")
    spec_soil=readline(STDIN)
    constants.spec_soil=parse(Int16,spec_soil)
    println("Insert material specific weight in Ib/ft3")
    spec_mat=readline(STDIN)
    constants.spec_mat=parse(Int16,spec_mat)
    println("Insert material shear strength in Ib/ft2 ")
    mat_shear=readline(STDIN)
    constants.mat_shear=parse(Float32,mat_shear)
    constants.mat_shear=(10^6)*constants.mat_shear
    println("Insert cohesion coefficient")
    cohe=readline(STDIN)
    constants.Cb=parse(Float32,cohe)
    println("Insert friction coeffiecient")
    frct=readline(STDIN)
    constants.Cf=parse(Float32,frct)
    println("Insert passive pressure coeffiecient")
    kp=readline(STDIN)
    constants.Kp=parse(Float32,kp)
    println("Insert base specific weight")
    spec_base=readline(STDIN)
    constants.spec_base=parse(Int16,spec_base)
    println("Allowable bearing Capacity of soil")
    Sb=readline(STDIN)
    constants.Sb=parse(Int16,Sb)
    println("Design Against Failure by sliding...............................")
    answer=design(barrier,constants)
    i = answer[1]
    coef= 0.5
    y= answer[2]
    while i < 1.5
        println("Changing base_width")
        coef= ch_base(barrier,coef)
        answer=design(barrier,constants)
        i= answer[1]
    end
    output_dimensions(barrier)
    println("designing against overturning moments..........................")
    answer = overturning(barrier,constants)
    ncoef=coef
    j = answer[1]
    while j < 1.5
        ncoef = ch_base(barrier,ncoef)
        answer = overturning(barrier,constants)
        j = answer[1]
    end
    println("Adjusted dimension for overturning moments.......................")
    output_dimensions(barrier)
```

```julia
end
function output_dimensions(barrier)
    height=barrier.height
    bwd=barrier.base_width
    blen=barrier.base_length
    bthick=barrier.base_thickness
    println("base_width: $bwd")
    println("base_length: $blen")
    println("base_thickness: $bthick")
    print("Wall props..")
    wallt = barrier.wall_thickness
    wallh = barrier.height
    println("Wall height: $wallh")
    println("Wall thickness: $wallt")
end
main(barrier,constants)

```