

```

diary on          % begin recording

% FEMA appeal for The Town of Harpswell, Cumberland county, Maine
% TRANSECT ID: CM-57-1
% calculation by SJH, Ransom Consulting, Inc. 16-Apr-2020
% 100-year wave runup using TAW methodology
% including berm and weighted average with foreshore if necessary
%
% chk nld 20200220
%
% This script assumes that the incident wave conditions provided
% as input in the configuration section below are the
% appropriate values located at the end of the foreshore
% or toe of the slope on which the run-up is being calculated
% the script does not attempt to apply a depth limit or any other
% transformation to the incident wave conditions other than
% conversion of the peak wave period to the spectral mean wave
% as recommended in the references below
%
% references:
%
% Van der Meer, J.W., 2002. Technical Report Wave Run-up and
% Wave Overtopping at Dikes. TAW Technical Advisory Committee on
% Flood Defence, The Netherlands.
%
% FEMA. 2007, Atlantic Ocean and Gulf of Mexico Coastal Guidelines Update
%
%
%-----
% CONFIG
%-----
fname='inpfiles/CM-57-1sta_ele_include.csv'; % file with station, elevation, include
% third column is 0 for excluded points
imgname='logfiles/CM-57-1-runup';
SWEL=8.9341; % 100-yr still water level including wave setup.
H0=3.4921; % significant wave height at toe of structure
Tp=4.6289; % peak period, 1/fma,
T0=Tp/1.1;

gamma_berm=1; % this may get changed automatically below
gamma_rough=0.8;
gamma_beta=1;
gamma_perm=1;

setupAtToe=-0.002769;
maxSetup=0.51165; % only used in case of berm/shallow foreshore weighted average

plotTitle='Iterative TAW for CM-57-1'

plotTitle =

Iterative TAW for CM-57-1

% END CONFIG
%-----

SWEL=SWEL+setupAtToe

SWEL =

8.931331

SWEL_fore=SWEL+maxSetup

SWEL_fore =

9.442981

% FIND WAVELENGTH USING DEEPWATER DISPERSION RELATION
% using English units
L0=32.15/(2*pi)*T0^2

L0 =

90.6089996880873

% Find Hb (Munk, 1949)
%Hb=H0/(3.3*(H0/L0)^(1/3))
%Db=-Hb/.78+SWEL; % depth at breaking

% The toe elevation here is only used to determine the average
% structure slope, it is not used to depth limit the wave height.
% Any depth limiting or other modification of the wave height

```

```

% to make it consistent with TAW guidance should be performed
% prior to the input of the significant wave height given above.
Ztoe=SWEL-1.5*H0

Ztoe =

        3.693181

% read the transect
[sta,dep,inc] = textread(fname,'%n%n%n%[^\\n]','delimiter',' ','headerlines',0);

% remove unselected points
k=find(inc==0);
sta(k)=[];
dep(k)=[];

sta_org=sta; % used for plotting purposes
dep_org=dep;

% initial guess at maximum run-up elevation to estimate slope
Z2=SWEL+1.5*H0

Z2 =

        14.169481

% determine station at the max runup and -1.5*H0 (i.e. the toe)
top_sta=-999;
toe_sta=-999;
for kk=1:length(sta)-1
    if ((Z2 > dep(kk)) & (Z2 <= dep(kk+1))) % here is the intersection of z2 with profile
        top_sta=interp1(dep(kk:kk+1),sta(kk:kk+1),Z2)
    end
    if ((Ztoe > dep(kk)) & (Ztoe <= dep(kk+1))) % here is the intersection of Ztoe with profile
        toe_sta=interp1(dep(kk:kk+1),sta(kk:kk+1),Ztoe)
    end
end

toe_sta =

        3.98328281478888

top_sta =

        20.1288737963701

% check to make sure we got them, if not extend the end slopes outward
S=diff(dep)./diff(sta);
if toe_sta== -999
    dy=dep(1)-Ztoe;
    toe_sta=sta(1)-dy/S(1)
end
if top_sta== -999
    dy=Z2-dep(end);
    top_sta=sta(end)+dy/S(end)
end

% just so the reader can tell the values aren't -999 anymore
top_sta

top_sta =

        20.1288737963701

toe_sta

toe_sta =

        3.98328281478888

% check for case where the toe of slope is below SWL-1.5*H0
% in this case interpolate setup from the setupAtToe(really setup as first station), and the max setup
% also un-include points seaward of SWL-1.5*H0
if Ztoe > dep(1)
    dd=SWEL_fore-dep;
    k=find(dd<0,1); % k is index of first land point
    staAtSWL=interp1(dep(k-1:k),sta(k-1:k),SWEL_fore);
    dsta=staAtSWL-sta(1);
    dsetup=maxSetup-setupAtToe;
    dsetdsta=dsetup/dsta;
    setup=setupAtToe+dsetdsta*(toe_sta-sta(1));
    sprintf('!!- Location of SWEL-1.5*H0 is %4.1f ft landward of toe of slope',dsta)
    sprintf('!!- Setup is interpolated between setup at toe of slope and max setup')

```

```

    sprintf('-!!-      setup is adjusted to %4.2f feet',setup)
    SWEL=SWEL-setupAtToe+setup;
    sprintf('-!!-      SWEL is adjusted to %4.2f feet',SWEL)
    k=find(dep < SWEL-1.5*H0)
    sta(k)=[];
    dep(k)=[];
else
    sprintf('-!!- The User has selected a starting point that is %4.2f feet above the elevation of SWEL-1.5H0\n',dep(1)
    sprintf('-!!- This may be reasonable for some cases.  However the user may want to consider:\n')
    sprintf('-!!-      1) Selecting a starting point that is at or below %4.2f feet elevation, or\n', Ztoe)
    sprintf('-!!-      2) Reducing the incident wave height to a depth limited condition.\n')
end

ans =

-!!- Location of SWEL-1.5*H0 is 85.8 ft landward of toe of slope

ans =

-!!- Setup is interpolated between setup at toe of slope and max setup

ans =

-!!-      setup is adjusted to 0.46 feet

ans =

-!!-      SWEL is adjusted to 9.39 feet

k =

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55

```

56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78

```
% now iterate converge on a runup elevation
tol=0.01; % convergence criteria
R2del=999;
R2_new=3*H0; %initial guess
R2=R2_new;
iter=0;
R2_all=[];
topStaAll=[];
Berm_Segs=[];
TAW_ALWAYS_VALID=1;
while(abs(R2del) > tol && iter <= 25)
    iter=iter+1;
    sprintf('!----- STARTING ITERATION %d -----!',iter)
    % elevation of toe of slope
    Ztoe
    % station of toe slope (relative to 0-NAVD88 shoreline)
    toe_sta
    % station of top of slope/extent of 2% run-up
    top_sta
    % elevation of top of slope/extent of 2% run-up
    Z2
    % incident significant wave height
    H0
    % incident spectral peak wave period
    Tp
    % incident spectral mean wave period
    T0

    R2=R2_new
    Z2=R2+SWEL
    % determine slope for this iteration
    top_sta=-999;
    for kk=1:length(sta)-1
        if ((Z2 > dep(kk)) & (Z2 <= dep(kk+1))) % here is the intersection of z2 with profile
            top_sta=interp1(dep(kk:kk+1),sta(kk:kk+1),Z2)
            break;
        end
    end
    if top_sta== -999
        dy=Z2-dep(end);
        top_sta=sta(end)+dy/S(end)
    end

    % get the length of the slope (not accounting for berm)
    Lslope=top_sta-toe_sta

    % loop over profile segments to determine berm factor
    % re-calculate influence of depth of berm based on this run-up elevation
    % check for berm, berm width, berm height
    berm_width=0;
    rdh_sum=0;
    Berm_Segs=[];
    Berm_Heights=[];
    for kk=1:length(sta)-1
        ddep=dep(kk+1)-dep(kk);
        dsta=sta(kk+1)-sta(kk);
        s=ddep/dsta;
        if (s < 1/15) % count it as a berm if slope is flatter than 1:15 (see TAW manual)
            sprintf('Berm Factor Calculation: Iteration %d, Profile Segment: %d',iter,kk)
            berm_width=berm_width+dsta; % tally the width of all berm segments
            % compute the rdh for this segment and weight it by the segment length
            dh=SWEL-(dep(kk)+dep(kk+1))/2
            if dh < 0
```

```

        chi=R2;
    else
        chi=2* H0;
    end
    if (dh <= R2 & dh >=-2*H0)
        rdh=(0.5-0.5*cos(3.14159*dh/chi)) ;
    else
        rdh=1;
    end
    rdh_sum=rdh_sum + rdh * dsta
    Berm_Segs=[Berm_Segs, kk];
    Berm_Heights=[Berm_Heights, (dep(kk)+dep(kk+1))/2];
end
if dep(kk) >= Z2 % jump out of loop if we reached limit of run-up for this iteration
    break
end
end
sprintf('!----- End Berm Factor Calculation, Iter: %d -----!',iter)
berm_width
rB=berm_width/Lslope
if (berm_width > 0)
    rdh_mean=rdh_sum/berm_width
else
    rdh_mean=1
end
gamma_berm=1- rB * (1-rdh_mean)
if gamma_berm > 1
    gamma_berm=1
end
if gamma_berm < 0.6
    gamma_berm =0.6
end
% Iribarren number
slope=(Z2-Ztoe)/(Lslope-berm_width)
Irb=(slope/(sqrt(H0/L0)))
% runup height
gamma_berm
gamma_perm
gamma_beta
gamma_rough
gamma=gamma_berm*gamma_perm*gamma_beta*gamma_rough
% check validity
TAW_VALID=1;
if (Irb*gamma_berm < 0.5 | Irb*gamma_berm > 10 )
    sprintf('!!! - - Iribarren number: %6.2f is outside the valid range (0.5-10), TAW NOT VALID - - !!!\n', Irb*gamma_berm)
    TAW_VALID=0;
else
    sprintf('!!! - - Iribarren number: %6.2f is in the valid range (0.5-10), TAW RECOMMENDED - - !!!\n', Irb*gamma_berm)
end
islope=1/slope;
if (slope < 1/8 | slope > 1)
    sprintf('!!! - - slope: 1:%3.1f V:H is outside the valid range (1:8 - 1:1), TAW NOT VALID - - !!!\n', islope)
    TAW_VALID=0;
else
    sprintf('!!! - - slope: 1:%3.1f V:H is in the valid range (1:8 - 1:1), TAW RECOMMENDED - - !!!\n', islope)
end
if TAW_VALID == 0
    TAW_ALWAYS_VALID=0;
end

if (Irb*gamma_berm < 1.8)
    R2_new=gamma*H0*1.77*Irb
else
    R2_new=gamma*H0*(4.3-(1.6/sqrt(Irb)))
end
% check to see if we need to evaluate a shallow foreshore
if berm_width > 0.25 * L0;
    disp('! Berm_width is greater than 1/4 wave length')
    disp('! Runup will be weighted average with foreshore calculation assuming depth limited wave height on berm')
    % do the foreshore calculation
    fore_H0=0.78*(SWEL_fore-min(Berm_Heights))
    % get upper slope
    fore_toe_sta=-999;
    fore_toe_dep=-999;
    for kk=length(dep)-1:-1:1
        ddep=dep(kk+1)-dep(kk);
        dsta=sta(kk+1)-sta(kk);
        s=ddep/dsta;
        if s < 1/15
            break
        end
        fore_toe_sta=sta(kk);
        fore_toe_dep=dep(kk);
    end
    upper_slope=(Z2-fore_toe_dep)/(top_sta-fore_toe_sta)
    fore_Irb=upper_slope/(sqrt(fore_H0/L0));
    fore_gamma=gamma_perm*gamma_beta*gamma_rough;
    if (fore_Irb < 1.8)
        fore_R2=fore_gamma*fore_H0*1.77*fore_Irb;
    else

```

```

        fore_R2=fore_gamma*fore_H0*(4.3-(1.6/sqrt(fore_Irb)));
    end
    if berm_width >= L0
        R2_new=fore_R2
        disp ('berm is wider than one wavelength, use full shallow foreshore solution');
    else
        w2=(berm_width-0.25*L0)/(0.75*L0)
        w1=1-w2
        R2_new=w2*fore_R2 + w1*R2_new
    end
    end % end berm width check
    % convergence criterion
    R2del=abs(R2-R2_new)
    R2_all(iter)=R2_new;
    % get the new top station (for plot purposes)
    Z2=R2_new+SWEI
    top_sta=-999;
    for kk=1:length(sta)-1
        if ((Z2 > dep(kk)) & (Z2 <= dep(kk+1))) % here is the intersection of z2 with profile
            top_sta=interp1(dep(kk:kk+1),sta(kk:kk+1),Z2)
            break;
        end
    end
    if top_sta== -999
        dy=Z2-dep(end);
        top_sta=sta(end)+dy/S(end);
    end
    topStaAll(iter)=top_sta;
end
ans =
!----- STARTING ITERATION 1 -----!
Ztoe =
        3.693181
toe_sta =
        3.98328281478888
top_sta =
        20.1288737963701
Z2 =
        14.169481
H0 =
        3.4921
Tp =
        4.6289
T0 =
        4.20809090909091
R2 =
        10.4763
Z2 =
        19.8691232594291
top_sta =
        36.2849262003111
Lslope =
        32.3016433855222
ans =
!----- End Berm Factor Calculation, Iter: 1 -----!
berm_width =
    0
rB =
    0
rdh_mean =
    1
gamma_berm =
    1
slope =
        0.500777686954445
Irb =
        2.55086398387982
gamma_berm =
    1
gamma_perm =
    1
gamma_beta =
    1
gamma_rough =
        0.8
gamma =
        0.8
ans =
!!! - - Iribaren number: 2.55 is in the valid range (0.5-10), TAW RECOMMENDED - - !!!
ans =
!!! - - slope: 1:2.0 V:H is in the valid range (1:8 - 1:1), TAW RECOMMENDED - - !!!
R2_new =
        9.21414563601113
R2del =
        1.26215436398887
Z2 =
        18.6069688954403
top_sta =
        32.707256490291
ans =

```

```

!----- STARTING ITERATION 2 -----!
Ztoe =
    3.693181
toe_sta =
    3.98328281478888
top_sta =
    32.707256490291
Z2 =
    18.6069688954403
H0 =
    3.4921
Tp =
    4.6289
T0 =
    4.20809090909091
R2 =
    9.21414563601113
Z2 =
    18.6069688954403
top_sta =
    32.707256490291
Lslope =
    28.7239736755021
ans =
!----- End Berm Factor Calculation, Iter: 2 -----!
berm_width =
    0
rB =
    0
rdh_mean =
    1
gamma_berm =
    1
slope =
    0.519210470804736
Irb =
    2.6447569940343
gamma_berm =
    1
gamma_perm =
    1
gamma_beta =
    1
gamma_rough =
    0.8
gamma =
    0.8
ans =
!!! - - Iribaren number: 2.64 is in the valid range (0.5-10), TAW RECOMMENDED - - !!!
ans =
!!! - - slope: 1:1.9 V:H is in the valid range (1:8 - 1:1), TAW RECOMMENDED - - !!!
R2_new =
    9.26427328867568
R2del =
    0.0501276526645462
Z2 =
    18.6570965481048
top_sta =
    32.8493469093386
ans =
!----- STARTING ITERATION 3 -----!
Ztoe =
    3.693181
toe_sta =
    3.98328281478888
top_sta =
    32.8493469093386
Z2 =
    18.6570965481048
H0 =
    3.4921
Tp =
    4.6289
T0 =
    4.20809090909091
R2 =
    9.26427328867568
Z2 =
    18.6570965481048
top_sta =
    32.8493469093386
Lslope =
    28.8660640945497
ans =
!----- End Berm Factor Calculation, Iter: 3 -----!
berm_width =
    0
rB =
    0
rdh_mean =

```

```

1
gamma_berm =
1
slope = 0.518391267305827
Irb = 2.64058413099492
gamma_berm =
1
gamma_perm =
1
gamma_beta =
1
gamma_rough = 0.8
gamma = 0.8
ans =
!!! - - Iribaren number: 2.64 is in the valid range (0.5-10), TAW RECOMMENDED - - !!!
ans =
!!! - - slope: 1:1.9 V:H is in the valid range (1:8 - 1:1), TAW RECOMMENDED - - !!!
R2_new = 9.26210240573921
R2del = 0.00217088293646839
Z2 = 18.6549256651684
top_sta = 32.8431933862879
% final 2% runup elevation
Z2=R2_new+SWEL
Z2 = 18.6549256651684
diary off
-1.000000e+00

```