```
diary on          % begin recording

% FEMA appeal for The Town of Harpswell, Cumberland county, Maine
% TRANSECT ID: CM-161-1
% calculation by SJH, Ransom Consulting, Inc. 20-Feb-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-161-1sta_ele_include.csv';   % file with station, elevation, include
                                      % third colunm is 0 for excluded points
imgname='logfiles/CM-161-1-runup';
SWEL=8.874;  % 100-yr still water level including wave setup.
H0=1.7143;    % significant wave height at toe of structure
Tp=2.1869;    % 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=-0.0060892;
maxSetup=0;   % only used in case of berm/shallow foreshore weighted average

plotTitle='Iterative TAW for CM-161-1'

plotTitle =

Iterative TAW for CM-161-1



% END CONFIG
%------------------------------

SWEL=SWEL+setupAtToe

SWEL =

               8.8679108

SWEL_fore=SWEL+maxSetup

SWEL_fore =

               8.8679108



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

L0 =

          20.224304141426


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

Ztoe =

                  6.2964608


% 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 =

                  11.4393608


% 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 =

          24.6418722967695


top_sta =

          30.6375354698812

% 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 =

          30.6375354698812

toe_sta

toe_sta =

          24.6418722967695



% 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  6.6 ft landward of toe of slope


ans =

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


ans =

-!!-        setup is adjusted to -0.00 feet


ans =

-!!-        SWEL is adjusted to 8.87 feet


k =

     1
     2
     3
     4




% 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('!!! - - Iribaren number: %6.2f is outside the valid range (0.5-10), TAW NOT VALID - - !!!\n', Irb*gam
        TAW_VALID=0;
    else
        sprintf('!!! - - Iribaren number: %6.2f is in the valid range (0.5-10), TAW RECOMMENDED - - !!!\n', Irb*gamma_
    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 ber
        % do the foreshore calculation
        fore_H0=0.78*(SWEL_fore-min(Berm_Heights))
```

```matlab
        % 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);
            upper_slope=(Z2-fore_toe_dep)/(top_sta-fore_toe_sta)
        end
        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+SWEL
    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 =
               6.2964608
toe_sta =
         24.6418722967695
top_sta =
         30.6375354698812
Z2 =
               11.4393608
H0 =
                    1.7143
Tp =
                    2.1869
T0 =
         1.98809090909091
R2 =
                    5.1429
Z2 =
         14.0141507213548
top_sta =
         33.6392607824415
Lslope =
         8.99738848567193
ans =
!------- End Berm Factor Calculation, Iter: 1 ---------!
berm_width =
     0
rB =
     0
rdh_mean =
     1
gamma_berm =
     1
slope =
         0.857769999999999
Irb =
         2.94621362954295
gamma_berm =
     1
```

```
gamma_perm =
     1
gamma_beta =
     1
gamma_rough =
                    0.8
gamma =
                    0.8
ans =
!!! - - Iribaren number:   2.95 is in the valid range (0.5-10), TAW RECOMMENDED - - !!!
ans =
!!! - - slope: 1:1.2 V:H is in the valid range (1:8 - 1:1), TAW RECOMMENDED - - !!!
R2_new =
          4.6187981354521
R2del =
        0.524101864547896
Z2 =
         13.4900488568069
top_sta =
         33.0282556592174
ans =
!---------------- STARTING ITERATION 2 -----------------!
Ztoe =
                 6.2964608
toe_sta =
         24.6418722967695
top_sta =
         33.0282556592174
Z2 =
         13.4900488568069
H0 =
                    1.7143
Tp =
                    2.1869
T0 =
         1.98809090909091
R2 =
          4.6187981354521
Z2 =
         13.4900488568069
top_sta =
         33.0282556592174
Lslope =
         8.38638336244787
ans =
!------- End Berm Factor Calculation, Iter: 2 ---------!
berm_width =
     0
rB =
     0
rdh_mean =
     1
gamma_berm =
     1
slope =
        0.857769999999999
Irb =
         2.94621362954295
gamma_berm =
     1
gamma_perm =
     1
gamma_beta =
     1
gamma_rough =
                    0.8
gamma =
                    0.8
ans =
!!! - - Iribaren number:   2.95 is in the valid range (0.5-10), TAW RECOMMENDED - - !!!
ans =
!!! - - slope: 1:1.2 V:H is in the valid range (1:8 - 1:1), TAW RECOMMENDED - - !!!
R2_new =
          4.6187981354521
R2del =
     0
Z2 =
         13.4900488568069
top_sta =
         33.0282556592174
% final 2% runup elevation
Z2=R2_new+SWEL
Z2 =
         13.4900488568069
diary off
-1.000000e+00
-1.000000e+00
```