```
%%   CLEANED UP GLACIER MODEL FOR CLEAR CREEK
%
%    UPDATED ON: March 31st, 2016
%
%    1D FTCS STAGGERED GRID NUMERICAL MODEL: CLEAR CREEK GLACIER
%    All of the code written in SI units
%
%    AUTHORS:    COLE C. PAZAR    and   ROBERT S. ANDERSON
%
%%  model basics
    clear global %
    clearvars    % clear variables each run
    figure(1)    % main figure for animation of the glacier
          clf

%% initialize

%  constants

step  = 200;        % this determines matrix sizes for the whole model

font  = 15;         % simply choose the whole graph's font size

rho_i = 917;        % density of glacial ice
g     = 9.81;       % gravitational acceleration near the surface
A     = 2.16e-16;   % glenn-nye flow law parameter [=] Pa-3 yr-1
slide = 0.5;        % ratio of sliding speed to internal deformation speed

%  set up distance array

    xmax  = 28512;    % changes based on what data you'd like to import
       dx = xmax/step;
        x = dx/2:dx:xmax-(dx/2);
    xedge = 0:dx:xmax;

% valley width as a function of distance downvalley (approximated)

    W_min1 = 1000; % meters
    phi = 5; % importance of tributary widening, ~ 15 km? ------ was 4 before
    m = 3; % controls the shape of the upstream expansion of width
    x_star1 = 1500; % how quickly does it shrink ?
    x_max1 = 28500;
    dx = x_max1/step;
    x1 = 0:dx:x_max1-1;
    shift = 1800;   % was 2000
    geom1 = (1 + phi.*(((x1+shift)/x_star1).^m).*exp(-((x1+shift)/x_star1)));

    W = W_min1 * geom1;

    Wedge = W(1:end-1)+0.5*diff(W); % interpolates valley width to cell edges
    Wedge = [Wedge(1) Wedge Wedge(end)]; % fixes the width boundary conditions

    % LOAD IN THE CLEAR CREEK TOPOGRAPHY
    load CC_new_profile.txt

    % without SMOOTHING FUNCTION

    zb = transpose((CC_new_profile(1:5:end)));

    % or: ADD A SMOOTHING FUNCTION
    % zb = transpose(smooth(CC_new_profile(1:5:end)));
```

```
    H = zeros(size(x)); % ice thickness array

    Q = zeros(size(x)); % pre-allocation discharge array

    z = zb+H; % update topography for the glacier

    zmax = max(zb);
    zmin = min(zb);

%  meteorology and mass balance

    ELA0      = 3400;      % SET THE AVERAGE ELA


    sigma_ELA = 200;       % uncertainty in the ELA, and the amplitude


    dbdz = 0.01;           % m/y/m, typically ~0.01,


    bcap = 0.60;            % m/yr, usually 1.25-2.00

    b0 = dbdz*(z-ELA0);
    b0 = min(b0,bcap);
    minzb = find(zb==min(zb));
    minb = b0(minzb);

%  set up the time array

     dt   = 0.0035;    % time step has to be small for glaciers
     tmax = 7000;      % max time interval of growth,
     tmin = 16.8;      % ka

     t = tmax:-dt:0;

     randomsize_t = 0.75*randn(size(t+1000));  % for randomized variables

%  plotting controls

    imax   = length(t);
    nplots = 40;
    tplot  = tmax/nplots;
    nframe = 0;
    border = 100; % for vertical border in the plotting sizes

%  new way to control the climate: guess-and-check fourier-type analysis

     big_period   = 6000;
     med_period   = 6000;
     small_period = 2500;

     big_shift   = -4000; % shift the periods
     med_shift   = 1000;
     small_shift = 750;

     big = (sigma_ELA/20)*sin(2*pi*(t+big_shift)/big_period);
     medium = (sigma_ELA/100).*sin(2*pi*(t+med_shift)/med_period);
     small = (sigma_ELA/10).*sin(2*pi*(t+small_shift)/small_period);

    ELA = ELA0*ones(1,length(t))+(sigma_ELA/(10)*randomsize_t ...
    + medium+big+small);
    % for the plotting of the average ELA that the random funciton
```

```
    % oscillates around:
    ELA_simple = ELA0*ones(1,length(t))+medium+big+small;

% define terminus and dam location

    term   = zeros(1,length(t)); % define the glacier terminus matrix size
    dam_ht = zeros(1,length(t)); % define the dam height matrix size

    zdambase = min(zb); % find the bottom of the arkansas river
       zbmax = max(zb); % find the highest part in the topography
       zbmin = zdambase;
    dambase  = find(zb==zbmin);
    xdambase = x(dambase);

    zlateral = zb; % tracks for max lateral moraine topography
    tracking_thickness = H; % tracks for max ice thickness

    % add a time counter

      tic

%% run the model

for i = 1:imax

b = dbdz*(z-ELA(i));     % local net balance calculated at cell centers
                         % at ice surface
b = min(b,bcap);

Hedge = H(1:end-1)+0.5*diff(H); % interpolates ice thickness to cell edges
S = abs(diff(z)/dx);            % slope of ice surface calc. at cell edges

Udef = (A/5).*((rho_i*g*S).^3).*(Hedge.^4); % mean defm speed
%Q = [ ];
Q = (A/5).*((rho_i*g*S).^3).*(Hedge.^5);    % internal deformation dischar.
Qsl = slide * Udef.*Hedge;                  % sliding discharge
Q = Q + Qsl;                                % update the new discharge
Q =[0 Q 0];                                 % takes care of the edge B.C.

  dHdt = b - (1./W).*(diff(Q.*Wedge)/dx); % continuity allowing W to vary

  H = H + (dHdt*dt);                      % updates ice thickness
  H = max(H,0);                           % takes care of the edge B.C.

     z = zb+H;                   % updates topography for the ice

     glacier = find(H>0);        % define the glacier

    % (approximates for the moraines)

    zlateral = max(zlateral,z); % find the maximum extent of the ice
    moraine_start = 20000;
    x_moraine = 19000:6:25000;
    z_moraine = 250:0.15:400;
    tracking_thickness = max(tracking_thickness,H);

% plotting for figure 1:

if rem(t(i),tplot)==0

    nframe=nframe+1%;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% PLOTS THE GLACIER AND TOPOGRAPHY
figure(1)
subplot('position',[0.07 0.55 0.74 0.4])
plot(x/1000,z,'c','linewidth',3) % updates the glacier w/ topography
    hold on
plot(x/1000,zlateral+5,'k:','linewidth',2) % lateral moraine heights
plot(x/1000,zb,'k','linewidth',3)
plot(x/1000,(max(ELA))*ones(size(x)),'r','linewidth',0.5) % max possible ELA
plot(x/1000,ELA0*ones(size(x)),'g','linewidth',1.5)
plot(x/1000,(min(ELA))*ones(size(x)),'r','linewidth',0.5) % min possible ELA
axis([0 xmax/1000 zmin-border zmax+border])
title('Clear Creek Valley paleoglacier, LGM numerical reconstruction')
xlabel('Horizontal distance [km]','fontname','arial','fontsize',font)
ylabel('Elevation [m]','fontname','arial','fontsize',font)
    ELA0_text = num2str(ELA0);
    ELA0_text2 =strcat('ELA center (',ELA0_text,' m)');
       legend('temperate valley glacier','approx. moraine extent'...
           ,'bed topography','ELA range')
set(gca,'fontsize',font,'fontname','arial')
    hold off

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% PLOTS MASS BALANCE
subplot('position',[0.855 0.55 0.10 0.4])
plot(b,z,'b','linewidth',2.5)
        hold on
plot(zeros(size(z)),z,'k--','linewidth',1.5)
plot(b0,zb,'b--','linewidth',2) % steady base
plot(b,ELA0*ones(size(x)),'g','linewidth',1.5)
plot(b,(max(ELA))*ones(size(x)),'r','linewidth',0.5)
plot(b,(min(ELA))*ones(size(x)),'r','linewidth',0.5)
        legend('b(z)','zero line','location','northwest')
axis([minb bcap+1 zmin-border zmax+border])
xlabel('b(z) [m/yr]','fontname','arial','fontsize',font)
title('Mass balance')
set(gca,'fontsize',font,'fontname','arial')
        % PLOT THE TIME (within the mass balance domain)
        shift_for_text = -40;
        time=num2str((t(i)/1000)+tmin);
        timetext=strcat('     time  =',time,'   ka');
        text(shift_for_text,3880,timetext,'fontsize',font)
        % PLOT THE ELA AVERAGE
        averageELA=num2str(round(ELA(i)));
        averageELAtext=strcat('     ELA  =',averageELA,'   m');
        text(shift_for_text,3800,averageELAtext,'fontsize',font)
            hold off

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   now define the analytic solution:
    Qanal = (cumsum(b)*dx)*m;      % steady state ice discharge w/out width
    Qanal = max(Qanal,0);         % fixes B.C.
% PLOTS THE ICE DISCHARGE
subplot('position',[0.06 0.1 0.18 0.35])
plot(xedge/1000,Q/1000,'c','linewidth',3)
    hold on
plot(x/1000,(Qanal/1000),'b:','linewidth',3)
axis([0 xmax/1000 0 40])
    legend('total ice discharge','integrated b(z)*m')
title('Q(x) (non-uniform width)')
xlabel('Horizontal distance [km]','fontname','arial','fontsize',font)
ylabel('Discharge [10^3 m^2/yr]','fontname','arial','fontsize',font)
```

```
    set(gca,'fontsize',font,'fontname','arial')
        hold off

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % PLOTS THE ICE THICKNESS
        subplot('position',[0.3 0.1 0.18 0.35])
    plot(x/1000,H,'c','linewidth',3)
         hold on
    plot(x/1000,tracking_thickness+5,'k:','linewidth',2) % lateral moraine heights
    plot(x_moraine/1000,fliplr(z_moraine),'k','linewidth',1.5)
       legend('total thickness','maximum extent','end lateral moraines')
    axis([0 xmax/1000 0 625])
    title('Thickness of the glacial ice')
    xlabel('Horizontal distance [km]','fontname','arial','fontsize',font)
    ylabel('Ice thickness [m]','fontname','arial','fontsize',font)
    set(gca,'fontsize',font,'fontname','arial')
        hold off

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % PRESCRIBED CLIMATE
         subplot('position',[0.55 0.1 0.4 0.35])
         plot((t/1000)+tmin,(ELA_simple),'b','linewidth',2.5)
         hold on
         plot((t/1000)+tmin,ELA0*ones(size(t)),'g','linewidth',1.5)
         plot((t(i)/1000)+tmin,ELA(i),'bo','linewidth',5)
         plot(x/1000,(min(ELA))*ones(size(x)),'r','linewidth',0.5)
         plot(x/1000,(max(ELA))*ones(size(x)),'r','linewidth',0.5)
%          plot((t/1000)+tmin,ELA,'k.','linewidth',1)
            grid on
            legend('mean ELA',ELA0_text2,'ELA ( t ) _i')
         ylabel('ELA [m]','fontname','arial','fontsize',font)
         xlabel('Time [ka]','fontname','arial','fontsize',font)
         title('Model ELA(t) approximated to the \delta^1^80 record')
         set(gca,'fontsize',font,'fontname','arial')
         axis([tmin (tmax/1000)+tmin min(ELA)-20 max(ELA)+20])

         pause(0.02)
end

end
```