```matlab
% Author: Siru Pauliina Jylhankangas
% Victoria University of Wellington (03/2023, MSc. in GPHS)

% %---------------------------------------------------------------------------------

% Applying real EQ's through VRH fracture medium
% Writing output and input parameters into a new datafile,

% Reference scripts:
% "SiruJ_MSc2022_SinglePlane_EQ_split_Misfit.m"
% "SiruJ_MSc2022_VRH_multifrac_elasticity.m"
% "SiruJ_MSc2022_VRH_crack_model_plotting_only_no_split.m"
% "SiruJ_MSc2023_VRH_EQ_Split_misfit_NEW.m"

% %---------------------------------------------------------------------------------

% Open new file for writing and save the results:

fileID = fopen('C:\Users\jylhansi\OneDrive - Victoria University of Wellington - STAFF\Desktop\Siru MSc\MSc_Siru (H)\MSAT CODE WORK\2023 FINAL Analysis\Test_data\TEST_RK_VRH_split_misfit.csv','w');
%fileID = fopen('C:\Users\jylhansi\OneDrive - Victoria University of Wellington - STAFF\Desktop\Siru MSc\MSc_Siru (H)\MSAT CODE WORK\2023 FINAL Analysis\Ceff_analysis_data\RK_small_inc_VRH_EQ_split_misfit_super400C_20MPa.csv', 'w');

fprintf(fileID,'%s , %s , %s , %s , %s , %s , %s , %s , %s , %s , %s , %s , %s , %s , %s , %s , %s , %s , %s , %s , %s , %s , %s , %s\n', 'event', 'station', 'slat', 'slon', 'spol' , 'baz', 'Inc' , 'inc', 'fastpol' , 'tlag' , 'tlagkm' , 'vp', 'modelpol', 'ModPol', 'vs1', 'vs2', 'MAXdVS', 'dtm', 'cd', 'Strike1', 'dip1', 'Strike2', 'dip2', 'misfit_lamda', 'misfit_simple');

 %s , %s , %s , %s\n', 'slat', 'slon', 'spol' , 'baz', 'Inc' , 'inc', 'fastpol' , 'tlag' , 'tlagkm' , 'vp', 'modelpol', 'ModPol', 'vs1', 'vs2', 'MAXdVS', 'dtm', 'cd', 'Strike1', 'dip1', 'Strike2', 'dip2', 'misfit_lamda', 'misfit_simple');
, %s , %s , %s , %s , %s , %s\n', 'slat', 'slon', 'spol' , 'baz', 'Inc' , 'inc', 'fastpol' , 'tlag' , 'tlagkm' , 'vp', 'modelpol', 'ModPol', 'vs1', 'vs2', 'MAXdVS', 'dtm', 'cd', 'Strike1', 'dip1', 'Strike2', 'dip2', 'misfit_lamda', 'misfit_simple');

% Read the data specs first, to see what is required
opts = detectImportOptions('RK_Stns_small_inc_all_km_final_short_Fmt_NO_lamdamax.csv');

% opts.VariableNames = ["eventnr","stnNr" "slat", "slon", "evla", "evlo", "distevstat", "depthkm", "baz", "spol", "Dspol", "tlag", "Dtlag", "fast", "Dfast", "anginc", "spolfast", "lambdamax", "lambda_min", "ttime", "maxfreq","rayp"];
% opts.VariableTypes =  ["double", "double", "double", "double", "double", "double", "double", "double", "double", "double", "double", "double", "double", "double", "double", "double", "double", "double", "double", "double", "double", "double"];

% Rotokawa lamda_max cell error:
opts.VariableNames = ["eventnr","stnNr" "slat", "slon", "evla", "evlo", "distevstat", "depthkm", "baz", "spol", "Dspol", "tlag", "Dtlag", "fast", "Dfast", "anginc", "spolfast", "lambda_min", "ttime", "maxfreq","rayp"];
opts.VariableTypes =  ["double", "double", "double", "double", "double", "double", "double", "double", "double", "double", "double", "double", "double", "double", "double", "double", "double", "double", "double", "double", "double"];
```

```
% define the variable data as 'a Table', it is easier to inspect in workspace:

RK_EQ_Data = readtable('RK_Stns_small_inc_all_km_final_short_Fmt_NO_lamdamax.csv',
opts)
RT_EQ_Data =
readtable("RK_Stns_small_inc_all_km_final_short_Fmt_NO_lamdamax.csv",'Format','aut
o');

% %--------------------------------------------------------------------------------

% Define rock andd fracture parameters for elastic medium/rock models:

                 vf1 = 0.80;     % used for VRH
                 vf2 = 0.20;     % used for VRH
                 VF = [vf1 vf2]; % used for VRH
                 % Crack density set for the models can be looped or held constant
                 % cd1 = 0.055 ;
                 % cd2 = 0.055 ;
                 ar  = 0.0001;   % 0.0001 is default aspect ratio (M. Savage)
                 rhm = 2700;     % deep rock density
                 vpm = 5.5;      % in TVZ wide range from 1.5 to 5.5km/s
                 vsm = vpm/1.75; % Vs as fraction of Vp might vary (Rawlinson
model of RK/TVZ)
                 vsc = 0 ;        % Infill Vs in fluid = 0km/s, but this might
vary with solid infill

%--------------------------------------------------------------------------------

% Read earthquake data, to obtain variables:

% Change numerical table into an 'double' array (possible to loop over):

EQ_Data = table2array(RK_EQ_Data);
% define dimensions of data array to write 'for- loop for ith variables of
% each variable array in datafile
[n, m] = size(EQ_Data);

% define variables from data file:
% evNr = EQ_Data(:,1);
% stnNr = EQ_Data(:,2);
% slat = EQ_Data(:,3);
% slon = EQ_Data(:,4);
% lat = EQ_Data(:,5);
% lon  = EQ_Data(:,6);
% dist = EQ_Data(:,7);
% z = EQ_Data(:,8);
% baz = EQ_Data(:,9);
% spol = EQ_Data(:,10);
% Dspol = EQ_Data(:,11);
% tlag = EQ_Data(:,12);
% Dtlag = EQ_Data(:,13);
% fast = EQ_Data(:,14);
% Dfast = EQ_Data(:,15);
% Inc = EQ_Data(:,16);
% spolfast = EQ_Data(:,17);
% ttime = EQ_Data(:,20);
% dfreq = EQ_Data(:,21);
% rayp = EQ_Data(:,22);
```

```matlab
% Rotokawa lambda_max error requires removing the column
% define new variables from data file:
evNr = EQ_Data(:,1);
stnNr = EQ_Data(:,2);
slat = EQ_Data(:,3);
slon = EQ_Data(:,4);
lat = EQ_Data(:,5);
lon  = EQ_Data(:,6);
dist = EQ_Data(:,7);
z = EQ_Data(:,8);
baz = EQ_Data(:,9);
spol = EQ_Data(:,10);
Dspol = EQ_Data(:,11);
tlag = EQ_Data(:,12);
Dtlag = EQ_Data(:,13);
fast = EQ_Data(:,14);
Dfast = EQ_Data(:,15);
Inc = EQ_Data(:,16);
spolfast = EQ_Data(:,17);
ttime = EQ_Data(:,19);
dfreq = EQ_Data(:,20);
rayp = EQ_Data(:,21);

%-----------------------------------------------------------------------------

% Transfer observed EQ incidence angle into correct angle for MSAT coordinate
system:

for Inc = Inc    ;
    inc = 90 - Inc ;
end

%-----------------------------

% Define the rotation of the matrix/fracture planes:

    %a = 0;  % null axis of rotation
    %b1 = 0:10:20 % corresponds 'dip' of fracture plane (80 deg dip)
    %c1 = -45; % corresponds azimuth of fracture plane ('plane' NE-SW azi 045)
    %b2 = 0:10:20 % 90deg dip
    %c2 = -90; % not sure where dip direction is on this set (need to check)

    for c1 = -25 ; %-65:10:-25 ; % crack plane strike rotation
        for b1 = -30:10:30 ; % crack plane dip rotation F1
            for b2 = -30:10:30 ; % crack plane dip rotation F2
                for c2 = -90;  % crack plane strike F2
                   for a = 0; % null axis of rotation
                       % 2 geothermal models in a loop (2 rhc and 2 vpc):
                        for rhc = 100.5:862.4:962.9 ;
                           for vpc = 0.5073:1.0569:1.5642 ;
                               for cd = 0.02:0.005:0.07 ; (0.055) or loop

%                  rhc1 = 962.9 ; % 100C, 10MPa
%                  rhc2 = 100.5; ; % 400C, 20Mpa Diff = 862.4
%                  vpc1 = 1.5642; % 100C, 10MPa
%                  vpc2 = 0.5073 ; % 400C, 20Mpa Diff = 1.0569
```

```
% Defining the model based on the infill:

    for  rhc = rhc  ;
        if rhc < 900 ;
          vpc = 0.5073 ;
          model = 'Supercritical' ;
        else
          model = 'Conventional' ;
          vpc = 1.5642 ;
      end
    end

   % annotating F1 correctly (RHR):

  for b1 = b1 ;
   if b1 < 0 ;
       strike1 = 90 - c1 - 180 ;
       dip1 = 90 + b1 ;
       dipdir1 = 'SE' ;
   else
       strike1 = 90 + c1 + 180 ;
       dip1 = 90 - b1 ;
       dipdir1 = 'NW' ;
   end
  end

  % annotating F2 correctly (RHR):

  for b2 = b2 ;
   if b2 < 0 ;
       strike2 = 90 - c2 - 180 ;
       dip2 = 90 + b2 ;
       dipdir2 = 'E' ;
   else
       strike2 = 90 + c2 + 180 ;
       dip2 = 90 - b2 ;
       dipdir2 = 'W' ;
   end
  end

  % Chaging negative angles from matrix rotation into
  % positive strike angle F1 for plotting annotation:

  for strike1 = strike1 ;
      if strike1 < 0 ;
          strike1 = -1 * strike1 ;
      else
          strike1 = strike1 ;
      end
  end

%-----------------------------------------
```

```
% Define 2 elasticity matrices and rotations for VRH model:

 [Ceff1,rh1] = MS_effective_medium('hudson',vpm,vsm,rhm,vpc,vsc,rhc,ar,cd);
 [ROTCeff1] = MS_rot3(Ceff1,a,b1,c1);

 [Ceff2,rh2] = MS_effective_medium('hudson',vpm,vsm,rhm,vpc,vsc,rhc,ar,cd);
 [ROTCeff2] = MS_rot3(Ceff2,a,b2,c2) ;

%-------------------------------------------------------------------------

% Calculatiang model plot grids and phase velocities:

% Define rad and degrees for incidents:

             cvect = 10 ;     % number of contours
             VPcvect = cvect ;
             AVScvect = cvect ;
             VScvect = cvect ;
             view_angle = [-90,90] ;
             rad = pi./180 ;
             deg = 180./pi ;

% Define X and Y:

% Set up inc-az grids (for ALL possible incident angles)
             [INC1,AZ1] = meshgrid([90:-6:0],[0:6:360]) ;
% generate X/Y matrices for plotting, (do not print in command window)
             [X1,Y1,Z1] = sph2cart(AZ1.*rad,INC1.*rad,ones(size(AZ1))) ;
% Invoke MS_phasevels to get wave velocities etc.
             [~,~,vs11,vs12,vp1, S1P1] = MS_phasevels(ROTCeff1,rh1,...
             reshape(INC1,61*16,1),reshape(AZ1,61*16,1));

             %----------------

% Set up inc-az grids (for ALL possible incident angles)
             [INC2,AZ2] = meshgrid([90:-6:0],[0:6:360]) ;
% generate X/Y matrices for plotting, (do not print in command window)
             [X2,Y2,Z2] = sph2cart(AZ2.*rad,INC2.*rad,ones(size(AZ2))) ;
% Invoke MS_phasevels to get wave velocities etc.
             [~,~,vs21,vs22,vp2, S1P2] = MS_phasevels(ROTCeff2,rh2,...
             reshape(INC2,61*16,1),reshape(AZ2,61*16,1));

% Reshape results back to grids:

             VS1_1 = reshape(vs11,61,16);
             VS2_1 = reshape(vs12,61,16);

             %------------------

             VS1_2 = reshape(vs21,61,16);
             VS2_2 = reshape(vs22,61,16);

% These are part of original code in MS_Plot (Phasevels)
% they are not required when retrieve th s-wave  anisotropy

             VP_1 =  reshape(vp1,61,16);
             VS1_1x = reshape(S1P1(:,1),61,16);
             VS1_1y = reshape(S1P1(:,2),61,16);
```

```
            VS1_1z = reshape(S1P1(:,3),61,16);
            %------------------

            VP_2 =  reshape(vp2,61,16);
            VS1_2x = reshape(S1P2(:,1),61,16);
            VS1_2y = reshape(S1P2(:,2),61,16);
            VS1_2z = reshape(S1P2(:,3),61,16);

% aVS data
            dVS_1 =  (VS1_1-VS2_1) ;
            VSmean_1 = (VS1_1+VS2_1)./2.0 ;
            AVS_1 = 100.0*(dVS_1./VSmean_1) ;
            %------------------
            dVS_2 =  (VS1_2-VS2_2) ;
            VSmean_2 = (VS1_2+VS2_2)./2.0 ;
            AVS_2 = 100.0*(dVS_2./VSmean_2) ;

%-------------------------------------------------------------------------
% Calculating VRH average for multiple fracture families:

            [Cav,rhav] = MS_VRH(VF, ROTCeff1, rh1, ROTCeff2, rh2);

% Define the average Effective medium as a combination of the two:

            C= Cav;
            rh = rhav;
%-------------------------------------------------------------------------
% Calculate Shear wave splitting for single fracture plane:

%  Loop over elements of other data arrays (rows in data csv)
% - this is n number of rows in the entire data file:

for i = 1:n;

[pol, avs, vs1, vs2, vp] = MS_phasevels(C, rh, inc(i), baz(i));

% convert negative fast pol into positive in 360' compass/mapview for
% plotting stereographic projections in stereonet:

            for pol = pol ;
                if pol < 0 ;
                    Pol = 360 + pol ;
                else
                    Pol = pol ;
                end
            end
%-------------------------------------------------------------------------

% define time delay for each element of split phase velocities:
A = 1;
% dist =
slw1 = A/vs1;
slw2 = A/vs2;
dtm = slw2-slw1; % model delay time

% Calculating the average time delay per kilometer on hypocentral distance:

dtd = tlag./dist;  % dt per kilometer of data
```

```
%------------------------------------------------------------------------

% Calculating the splitting misfit between model and data polarisations:

% - define the misfit parameters

tlag1 = dtm ; % this is the model delay time per unit kilometer

tlag2 = dtd ; % this is the data time delay per kilometer (average) % tlag2 = tlag
% this is Stefans observed time delay over the distance

fast1 = pol ; % this is model split polarisation of fast phase
fast2 = fast ; % this is Stefans fast phase polarisation angle

misfit1(i) = MS_splitting_misfit(fast1, tlag1, fast2(i), tlag2(i), spol(i),
dfreq(i), 'mode', 'lam2');
misfit2(i) = MS_splitting_misfit(fast1, tlag1, fast2(i), tlag2(i), spol(i),
dfreq(i), 'mode', 'simple');

misfit_lam = misfit1' ;
misfit_simp = misfit2' ;

%------------------------------------------------------------------------
% Printing data in new CSV file:

Dparams = [evNr(i), stnNr(i), slat(i), slon(i), spol(i), baz(i), Inc(i), inc(i),
fast(i), tlag(i), dtd(i), vp, pol, Pol vs1, vs2, avs, dtm, cd, strike1, dip1,
strike2, dip2, misfit_lam(i), misfit_simp(i)]';

fprintf(fileID, '%f , %f , %f,  %f , %f , %f , %f , %f , %f , %f , %f , %f , %f ,
%f , %f , %f , %f , %f , %f , %f , %f , %f , %f , %f , %f\n', Dparams);


% END all loops:

end

                               end
                           end
                        end
                   end
                end
            end
        end
    end

%--------------------------------------------------------------------------------

% CLOSE the new data file in writing:

fclose(fileID)

% END OF SCRIPT

%--------------------------------------------------------------------------------
%--------------------------------------------------------------------------------
```