Table of Contents

Midterm_W19_KEY.m

JMA 2.7.19

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Make a movie of the theoretical Insight temperature profile over
% time using the following estimated values for Martian conditions:
%
%
%   Thermal conductivity of Mars Rock ~ 0.03 W/m K.
%   Density of sandstone ~ 1300 kg/m^3.
%   Heat capacity        ~ 650 J/kg K.
%
%   Estimated outwards heat flux ~ 0.025 W/m^2.
%
%   Plot a thin straight magenta line in the back ground at fixed
%   temperature T_o.
%
% SOLUTION:
%
%
%   T(t,z) = [T_o + T(z)_HFlux] + {T_Season(t,z) + T_Day(t,z)}
%
%   Square brackets = Steady Temp field
%   Curly  brackets = Oscillatory Temp field
%
%   with    T_o ~ 224 K ~ -42 deg C  = the mean temperature (constant)
%
%           A_Y ~ 11 K = amplitude of the yearly (seasonal)
%                        surface thermal wave
%           Per_Y = 668.6 Sols = 5.93e7 s = the seasonal period;
%                       = 1.881 Earth years
%
%           A_D ~ 36 K = amplitude of the daily surface thermal wave
%           Per_D = 24 hrs. 37 min = 88,643 s = the daily period = 1
 Sol.
%                 = 1.026 Earth days
%
%               Temp. Estimates: Gale Crater also near equator
%                   Data from Climate of Mars (Wikipedia) Montly Aves
```

```
%                   which agree well with Martin-Torres et al.
%                   Nat Geo 2015 (or are the same thing!)
%
%
%           z = ~0.15:4.5 m is depth over which the HP3 takes thermal
%               data via 14 thermistors in a vertical chain
%           delta = \sqrt{kappa*Per/ pi} = the skin depth,
%               where kappa = thermal diffusivity
%               and   Per = oscillation period
%           t is time
%
%       T_SkDpth  = A*exp(-z/delta)*cos([2*pi/Per]*t - z/delta)
%
%
%
%
%   Steady Conduction Solution
%
%   |q| = K (d(T_HF(z)) / dz)
%
%   with pre-InSight estimates of mean q ~ 30 mW/m^2.
%
%
%   T_HF(z) = (q /K)*(z)
%
%   where q_z ~ 0.030 W/m^2 and
%   K   ~ 0.03 W/mK     (Spohn, bottom page 96)
%   rho ~ 1300 kg/m^3   (Spohn, bottom page 96)
%   C   ~ 650 J/kg K    (Typical of Sandstones...)
%
%   (Therefore, kappa = K/rho C = 3.5 \times 10^-8 m^2/s)
%
%   At zmax = 4.5 m, we get T_HF(zmax) = T_o + 4.5 K.
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

INITIAL SET UP:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    clc;
    clear;
    close all;
    format long

    %Set linewidths, fontsizes, markersizes in plots:
    Linewid = 1.01;             %Linewidth size
    fsize   = 21;               %Font size
    axisize = 15;               %Axis label text size
```

```
lgdsize = 15;                %Legend text size
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

DATA GENERATION CODE BLOCK !!!!

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Calculate T2D(t,z): 2D matrix z rows by t columns
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

          %%%%%%%%%%%%%%%%%%%%%%%%
          %MATERIAL ESTIMATES:
          %%%%%%%%%%%%%%%%%%%%%%%%

              K   = 0.03;             %W/(m K)  (Spohn, bottom page 96)
              rho = 1300;             %kg/m^3   (Spohn, bottom page 96)
              C   = 650;              %J/kg K   (Typical of
 Sandstones...)

              kappa = K/(rho*C);      %~3e-8 m^2/s (rather low...)

          %%%%%%%%%%%%%%%%%%%%%%%%
          %THERMAL INFO & OSCILLATION PERIODS
          %%%%%%%%%%%%%%%%%%%%%%%%

              %Surface Mean Temp:
                    T_o     = -42;                  %deg C

              %DIURNAL (Sols)
                    A_D       = 36;                 %deg C
                    Per_D     = 88642.663;         %seconds

                    %Therefore,
                    delta_D = sqrt(kappa*Per_D/pi); %m

              %YEARLY (Seasonal)
                    A_Y       = 11;                 %deg C
                    Per_Y     = 668.5991*Per_D;     %seconds

                    %Therefore,
                    delta_Y = sqrt(kappa*Per_Y/pi);  %m


              %Estimated Heat Flux:
                    q = 0.031;                      %W/m^2

            %%%%%%%%%%%%%%%%%%%%%%%%
            %INSIGHT NUMBERS:
            %%%%%%%%%%%%%%%%%%%%%%%%
```

```
            zmin = 0.50;                %meters, 1st  thermistor
depth
            zmax = 4.5;                 %meters, last thermistor
depth
            step = (zmax - zmin)/14;    %even thermistor spacing
            z = zmin: step :zmax;       %meteres

            tmax = Per_Y;              %seconds, Mars year
            t = 0: Per_D/1 :tmax;      %seconds



            %GENERATES 2D ARRAYS (length(t) x length(z))
            [Time2D, Z2D] = meshgrid(t,z);
                    %2D arrays with
                    %depth increasing down columns (Z2D),
                    %and time increasing along rows (Time2D).


   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   %CALCULATE STEADY TEMPERATURE FIELD (fcn of z only):
   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

   T1D_Steady = T_o + (q/K).*z;
   T2D_Steady = T_o + (q/K).*Z2D;

   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   %CALCULATE OSCILLATORY TEMPERATURE FIELD (fcn of t and z):
   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

   T2D_Osc = A_Y*exp(-Z2D/delta_Y).*cos(2*pi*Time2D/Per_Y  ...
                                    - Z2D/delta_Y) +    ...
             A_D*exp(-Z2D/delta_D).*cos(2*pi*Time2D/Per_D  ...
                                    - Z2D/delta_D)      ;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Calculate T2D(t,z): 2D matrix z rows by t columns
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

   T2D = T2D_Steady + T2D_Osc;

   %ADD +/-6.5 milliKelvin NOISE!
   noise = 0.0065*randn(size(T2D));

   %Commented out for Exam; Include for Prob 2 of HW4.
   %T2D = T2D + noise;


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Save Info to .mat File
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    save Midterm.mat
```

%%%

END OF DATA GENERATION BLOCK !!!!

%%%

%%%

T1D_mean Calculation (mean over the 1 year of data sampling)

%%%

```
T1D_year = mean(T2D, 2);
    %takes mean over the second array dimension, j, of T2D(i,j)
        %where i controls the depth z location
        %and   j controls the time  t location
```

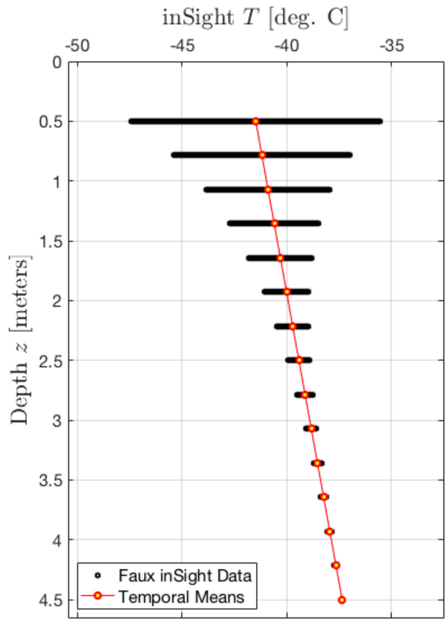
%%%

PLOT Temperature vs. Depth z

%%%

```
%SET UP FIGURE

    %Figure ID
        f1 = figure(1);
        %White background
            set(f1,'color','white');
        %Custom figure size
            set(f1, 'Position', [600, 600, 1000, 850]);

  %OVERTITLE:
  title_cell = {['EPSS 171 Midterm; J. Aurnou ' date], [' ']};
  sgtitle(title_cell, 'fontsize',0.825*fsize, ...
          'interpreter','latex');

  % SUBPLOT!!!
  m = 2;
  subplot(1,m,1)

 %DEFINE AXES RANGES:
     xmin = min(min(T2D)) - 3; xmax = max(max(T2D)) + 3;
     ymin = 0; ymax = 4.65;
```

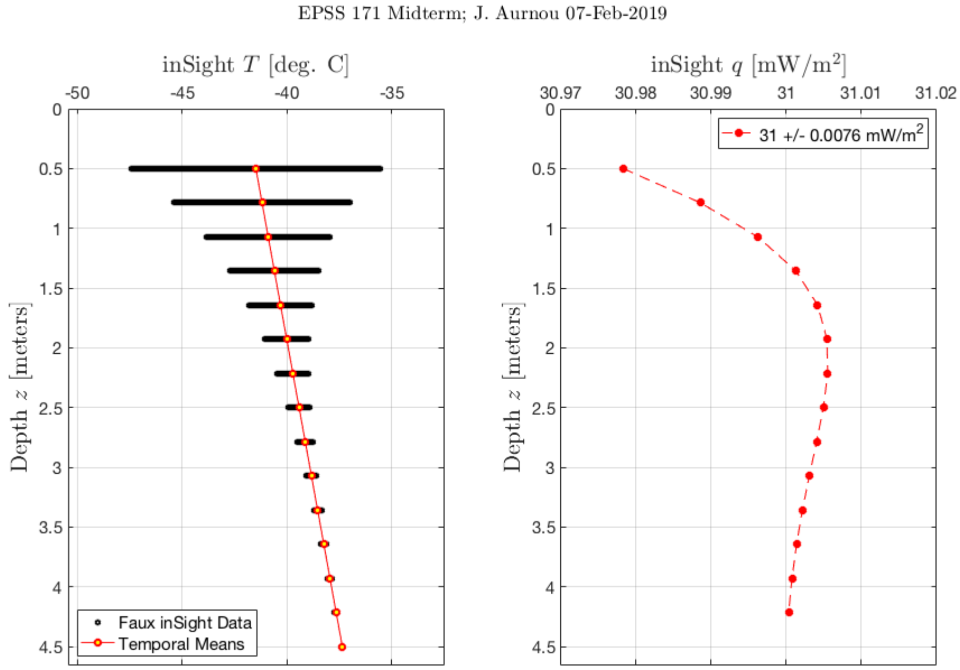
```
%PLOT DAILY MEASUREMENTS IN A LOOP:
   for i = 1: length(t)     %YEAR DATASET (669 simulated InSight days)

        if i == 1
            hndl(1) = plot(T2D(:,i), z, ['ok'], 'linewidth', 0.125,...
                      'markersize', 4);
            set(gca, 'Ydir', 'reverse');
            set(gca, 'XAxisLocation', 'top')
            set(gca, 'fontsize', axisize);
            axis([xmin xmax ymin ymax]);
            grid on
            hold on
        else
            plot(T2D(:,i), z, ['ok'],'linewidth',0.125,'markersize',
 4)
            %(:,i) means plot at all z values, at i-th step in time...
        end

    end

%PLOT STEADY PROFILE ON TOP IN RED & YELLOW:
hndl(2) = plot(T1D_year, z, '-or', 'linewidth', 1.01, ...
     'markerfacecolor', 'y');


 %LABEL AXES AND LEGEND (POST LOOP)

    % SET AXIS LABELS (fancy text interpreter not on test.)
        xlabel('inSight $T$   [deg. C]','interpreter','latex',...
            'fontsize', fsize);
        ylabel('Depth $z$   [meters]','interpreter','latex',...
            'fontsize', fsize);


lgd_cell = {['Faux inSight Data'], ['Temporal Means']};
legend(hndl, lgd_cell, 'location', 'southwest', 'fontsize', lgdsize)



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```



PLOT Heat Flux q vs. Depth z

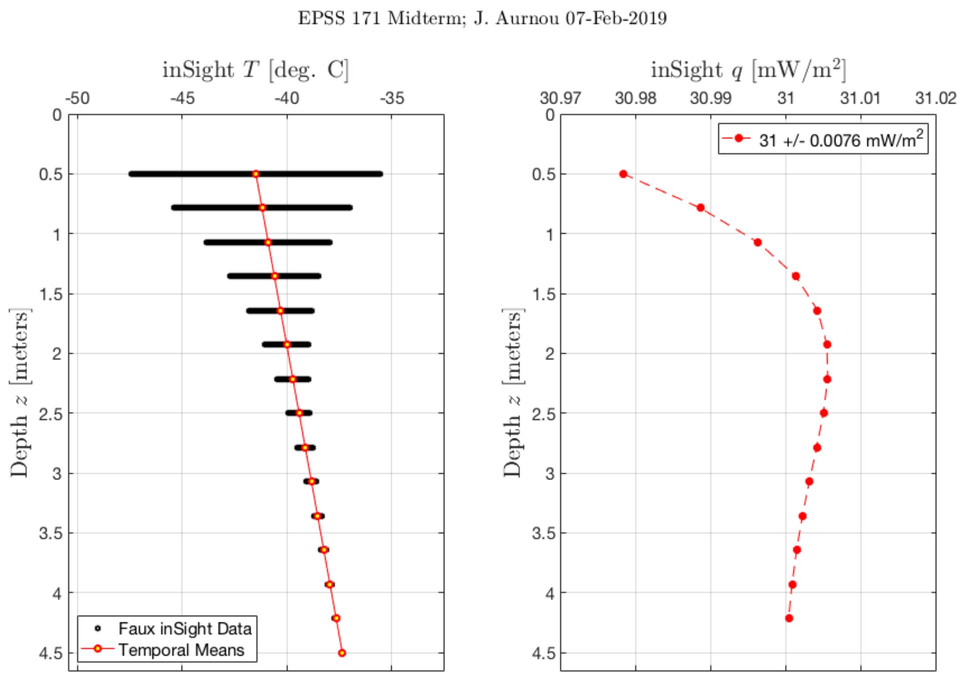
```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %CALCULATE heat flux q:
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    q_year = [1:1:length(z)-1];
        %one less than length of z, because we use zmax as our
        %anchor point in dT/dz, so not possible to calculate
        %dT/dz at zmax...

    for j = 1: (length(z) - 1)

        %Calculate |q| between bottom thermistor (z = 4.5 m) and
        %higher thermistors using Fourier's Law (q = K grad T)

        %q = K * (delta T) / (delta z)
        q_year(j) = K * (T1D_year(end) - T1D_year(j)) ./ ...
                                 (z(end) -      z(j));
                  %q_year(1) is estimated HF on first thermistor
                  %below the planet's surface.  Then higher indices
                  %correspond to q_year estimates on successively
                  %deeper situated thermistors.
                  %
                  %Again, no value at z = 4.5, since, for this
                  %formulation, that would correspond to
                  %K * 0 / 0 ---> Which does not exist / NaN!!!
```

```
    end

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


      %SUBPLOT: PLOT q vs. z:
      subplot(1,m,2)

      %Heat flux vs. z:
%     plot(1000*q_year, fliplr(z(1:end-1)),'--or', 'linewidth',
 1.01, ...
      plot(1000*q_year, (z(1:end-1)),'--or', 'linewidth', 1.01, ...
           'markerfacecolor', 'r')
        set(gca, 'Ydir', 'reverse');
        set(gca, 'XAxisLocation', 'top')
        set(gca, 'fontsize', axisize);
        %axis([24.98 25.02 ymin ymax]);
        ylim([ymin ymax]);
        xlim([30.97 31.02])
        grid on
        hold on


    % SET AXIS LABELS (fancy text interpreter not on test.)
        xlabel('inSight $q$   [mW/m$^2$]','interpreter','latex',...
            'fontsize', fsize);
        ylabel('Depth $z$   [meters]','interpreter','latex',...
            'fontsize', fsize);


    %legend
    leg_cell = {[n2s(1000*mean(q_year),5) ' +/- ' ...
                 n2s(1000*std(q_year), 2) ' mW/m^2']};
    legend(leg_cell, 'location', 'northeast', 'fontsize', lgdsize);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```



PRINT OUT PLOT TO PNG

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%OUTPUT PNG:
     %Output filename string
        figname1 = 'Midterm_W19_Aurnou';

     %Prints custom fig size
        f1.PaperPositionMode = 'auto';

     %PNG figure format
        print(f1,'-dpng','-r225',[figname1 '.png']);



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Midterm Winter 2019.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```



Published with MATLAB® R2018b