PBU Figure Plot

We plot pressure and derivative of PBU data of a selected Well

Other functions required:

```
pbu_NaN_removed=remove_nan(pbu)

pbu_interp=interpolate_data(pbu_NaN_removed, sampling_rate)

[xlv,ylv]=get_limit(t_NaN_removed,X_NaN_removed)
```

Contents

clear previous figure, data

```
clear all;
close all;
clc;
```

Figure display and save options

```
make_figure_visible='off';      % 'on' if want to display figure in matlab while running, 'off'
 otherwise
ghostscript_available=false(1,1); % true(1,1) if ghostscript has been installed, false(1,1) ot
herwise
```

Sampling rate

```
sampling_rate=0.01;
```

Directory creation for raw input data and output result

```
code_dir=pwd;                                   % directory where all the code exists

home_dir=strcat(code_dir,'/..');                % home directory, include code, data, result fold
er

pdf_maker_dir=strcat(code_dir,'/export_fig');% path directory for export_fig toolbox
addpath(genpath(pdf_maker_dir))

cd (home_dir)

if ~(exist ('DATA','dir'))                       % checks if there is already a DATA folder
    mkdir DATA;                                  % if not, creates one
end

if ~(exist ('RESULT','dir'))                     % checks if there is already a RESULT folder
    mkdir RESULT;                                % if not, creates one
end
```

Select Well PBU CSV data folder

```
data_path=strcat(home_dir,'/DATA');             % path to the Well PBU data folder
cur_data_path = uigetdir(data_path);

[~,cur_data_folder_name] = fileparts(cur_data_path);   % name of the selected Well PBU data fo
lder

cd(cur_data_path);

disp('Running...')
```

Read pressure derivative

```
RM_fileList = dir('*_RM.csv');

L=length(RM_fileList);

PBU=cell(1,L);

for k=1:L

    filenames=RM_fileList(k).name;

    val=csvread(filenames,1,0);
    PBU{1,k}=val;

end
```

Read delta pressure

```
log10dp_fileList = dir('*_log10dp.csv');

L=length(log10dp_fileList);

log10dp=cell(1,L);

for k=1:L

    filenames=log10dp_fileList(k).name;

    val=csvread(filenames,1,0);
    log10dp{1,k}=val;

end

cd (home_dir);
cd ('RESULT');

cur_data_result_folder_name=strcat(cur_data_folder_name,'_RESULT');

if ~(exist (cur_data_result_folder_name,'dir'))           % checks if there is already a Resul
t folder for the selected well
    mkdir (cur_data_result_folder_name);                  % if not, creates one
end

% cd (cur_data_result_folder_name);
%
% pbu_file_name=strcat(cur_data_folder_name,'.mat');
% save(pbu_file_name,'PBU','log10dp');                    % saves pbu pressure and derivative da
ta in a mat file

cd (code_dir);

%  creating a cell variable 'file_name_cell'
% to store the individual PBU figure name
% in order to append them in future in a single file
file_name_cell=cell(1,length(PBU));
```

Run a for loop to go through all the pbu data one at a time and plot figure

```
for pbu_no=1:length(PBU)
```

Loads derivative data

loading both Time axis value and Derivative value in a Matrix. Size of matrix *pbu* is 2 by 'PBU-Time Series length'. First row contains Time axis value Second row contains Derivative value

```
    pbu=PBU{1,pbu_no};
```

Nan Removal

Removes the data points where derivative is NaN size of input and output matrix is 2 by 'PBU-Time Series length'. First row contains Time axis value Second row contains Derivative value

```
pbu_NaN_removed=remove_nan(pbu);

t_NaN_removed=pbu_NaN_removed(:,1);    % time axis values of NaN removed data

X_NaN_removed=pbu_NaN_removed(:,2);    % derivative values of NaN removed data
```

Linear interpolation

Performs linear interpolation on the NaN removed Data to make the data uniformly sampled using given sampling rate size of input and output matrix is 2 by 'PBU-Time Series length'. First row contains Time axis value Second row contains Derivative value

```
pbu_interp=interpolate_data(pbu_NaN_removed, sampling_rate);

t_interp=pbu_interp(:,1);                % time axis values of interpolated data

X_interp=pbu_interp(:,2);                % derivative values of interpolated data
```

Loads Pressure Data

loading both Time axis value and pressure value in a Matrix. size of the matrix *dp_val* is 2 by 'PBU-Time Series length'. First row contains Time axis value Second row contains pressure value

```
dp_val=log10dp{pbu_no};

dp_time=dp_val(:,1);                     % time axis value of pressur data

dp_derv=dp_val(:,2);                     % pressure value
```

Obtains plotting axis limit

obtains the minimum, maximum of time axis value and derivative value for plot axis limit

```
[xlv,ylv]=get_limit(t_NaN_removed,X_NaN_removed);
```

Plot figure

```
figure('visible','off'),...
    plot (dp_time,dp_derv,'.b');
xlim(xlv),ylim(ylv),grid on,GridLineStyle=':';
set(gca,'XTick', xlv(1)+.2:xlv(2)-.2,'YTick', ylv(1)+.2:ylv(2)-.2);
xlabel('Log(delta time)','FontSize',8),...
    ylabel('Log(delta pressure) & Log(derivative)','FontSize',8);
hold on;
```

```
    plot (t_NaN_removed,X_NaN_removed,'.k');

    title_str=strcat('PBU: ',num2str(pbu_no));

    title (title_str);

    hold off;
```

Save figure

```
    fig_save_name=strcat(home_dir,'/RESULT/',cur_data_result_folder_name,...
        '/',cur_data_folder_name,'_PLOT_',num2str(pbu_no),'.pdf');

    set(gcf, 'Color', 'w');

    print(gcf,'-dpdf',fig_save_name,'-bestfit');

    file_name_cell{1,pbu_no}=fig_save_name;
```

```
end
```

Append all the pdf in a single pdf

Has dependency on the availability of Ghostscript

```
if (ghostscript_available)

    final_output_name=strcat(home_dir,'/RESULT/',cur_data_result_folder_name,...
        '/',cur_data_folder_name,'_PLOT_ALL.pdf');

    if exist(final_output_name,'file')
        delete (final_output_name);
    end

    append_pdfs(final_output_name, file_name_cell{:});

    for w=1:length(file_name_cell)
        delete (file_name_cell{w})
    end


end

clc;
disp('Finished')
```

Published with MATLAB® R2018a