Experimental investigation of the effect of yaw angle on the inflow of a two-bladed propeller  
The data provided in this data object contains two formats of the same experimental data.   
Due to usage of data structures**, it is recommended that the .mat files be used** instead of the CSV ones, which will be significantly harder to process.   
  
**Note**: If **CSV** format is used, the parameters will **not** have the same locations in the table for each file due to varying number of reflection points removed (usually between 0 and 10).  
As such, the best approach would be to use a code scanner for the start and end of each parameter using the first row of the CSV file.  
  
The file name is assembled using the following information:  
  
Example: **Vinf**9**P**12**RPM**2981K0**I**0.C  
**Vinf**(inity) (Experimental Freestream velocity) = 9 [m/s]  
**P**(itch) angle at 0.7R position = 12 [Deg]  
**RPM** of the motor = 2981 [Rev/min]  
K- not relevant for this project  
**I**(ncline) angle (Yaw) = 0 [Deg]  
  
Each .mat file contains the following parameters:  
PointStructure2.**AVU**- 2D array of propeller U velocity component (normalised by Vinf) in the experimental plane. (Interpolated data)  
PointStructure2.**AVV**- 2D array of propeller V velocity component (normalised by Vinf) in the experimental plane. (Interpolated data)  
PointStructure2.**AVW**- 2D array of propeller W velocity component (normalised by Vinf) in the experimental plane. (Interpolated data)  
  
PointStructure2.**XGnd**- 2D array of non-dimensionalised X coordinates for the velocity components AVU,AVV and AVW.  
PointStructure2.**YGnd**- 2D array of non-dimensionalised Y coordinates for the velocity components AVU,AVV and AVW.  
  
PointStructure2.**xqnd**- 1D array of non-dimensionalised X coordinates for the velocity components AVU,AVV and AVW.  
PointStructure2.**yqnd**- 1D array of non-dimensionalised Y coordinates for the velocity components AVU,AVV and AVW.

PointStructure2.**Azim**- 2D array of azimuthal position coordinates for the velocity components AVU,AVV and AVW. [**Radians**]  
PointStructure2.**Rad**- 2D array of non-dimensionalised radial position coordinates for the velocity components AVU,AVV and AVW.  
PointStructure2.**AzimD**- 2D array of azimuthal position coordinates for the velocity components AVU,AVV and AVW. [**Deg**]  
  
PointStructure2.**MeaningfulMeanU**- 1D array of propeller U velocity component(normalised by Vinf) in the experimental plane. (Non-interpolated measurement points)  
PointStructure2.**MeaningfulMeanV**- 1D array of propeller V velocity component(normalised by Vinf) in the experimental plane. (Non-interpolated measurement points)  
PointStructure2.**MeaningfulMeanW**- 1D array of propeller W velocity component(normalised by Vinf) in the experimental plane. (Non-interpolated measurement points)  
PointStructure2.**ProcCoords**-3 1D arrays of coordinates (**X,Y,Z in rows 1,2,3**) for the velocity components MeaninfulMeanU, V and W. (Dimensional coordinates in 3D traverse coordinate axis)  
  
PointStructure2.**PCAzim**- 1D array of azimuthal position coordinates for the velocity components MeaninfulMeanU, V and W. [**Radians**]  
PointStructure2.**PCRad**- 1D array of non-dimensionalised radial position coordinates for the velocity components AVU,AVV and AVW.  
PointStructure2.**PCAzimD**- 1D array of azimuthal position coordinates for the velocity components MeaninfulMeanU, V and W. [**Deg**]  
  
  
**Each .csv contains the same parameters without the main structure name (PointStructure2.).   
The parameters are positioned longitudinally next to each other on the table, with the start of each parameter being marked with its name on the first row.**The transformation matrix used to achieve radial and tangential inflow factors, as seen from the inflow perspective (Looking at the propeller from upstream):

PointStructure2.AzimDm=PointStructure2.AzimD+180; % 0 to 360

**Radial**   
U\_rm =PointStructure2.AVW.\*cosd(PointStructure2.AzimDm) PointStructure2.AVV.\*sind(PointStructure2.AzimDm);   
**Tangential**

U\_thetam =-PointStructure2.AVW.\*sind(PointStructure2.AzimDm) + PointStructure2.AVV.\*cosd(PointStructure2.AzimDm);  
  
  
Surfaces can consequently be directly plotted as follows:  
s1=surf(PointStructure2.xqnd,PointStructure2.yqnd,PointStructure2.AVU)

Provided processed samples can be used for orientation checks.