In [1]:
```
#Importing libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import re
```

In [2]:
```
#Reading data
data=open("D:/Data_Analytics_Portfolio/Data_Extraction/FRF_SOL111.pch","r").re
```

In [3]:
```
#printing first 2000 characters
print(data[:2000])
```

```
$TITLE   =
1
$SUBTITLE=
2
$LABEL   = 1X
3
$DISPLACEMENTS
4
$MAGNITUDE-PHASE OUTPUT
5
$SUBCASE ID =           1
6
$FREQUENCY =   1.0000000E+00
7
      1001       G      1.450114E-04      3.111762E-04      1.670763E-04
8
-CONT-                  1.888825E-06      1.955736E-06      1.914576E-06
9
-CONT-                  3.565663E+02      1.765663E+02      1.765663E+02
10
-CONT-                  1.765662E+02      3.565664E+02      1.765663E+02
11
$TITLE   =
12
$SUBTITLE=
13
$LABEL   = 1X
14
$DISPLACEMENTS
15
$MAGNITUDE-PHASE OUTPUT
16
$SUBCASE ID =           1
17
$FREQUENCY =   2.0000000E+00
18
      1001       G      1.450185E-04      3.112022E-04      1.670868E-04
19
-CONT-                  1.889065E-06      1.955763E-06      1.914738E-06
20
-CONT-                  3.565661E+02      1.765660E+02      1.765661E+02
21
-CONT-                  1.765658E+02      3.565663E+02      1.765660E+02
22
$TITLE   =
23
$SUBTITLE=
24
$LABEL   = 1X
```

In [4]:
```
#Regex
Response=r'\$DISPLACEMENTS|\$VELOCITY|\$ACCELERATION'
Subcase=r'\$SUBCASE ID+\s+\=+\s+(\d+)'
Frequency=r'\$FREQUENCY\s*=\s*([\d.E+-]+)'
Values = r'\d+\s+G\s+([\d.E+-]+)\s+([\d.E+-]+)\s+([\d.E+-]+)'
```

In [5]:
```
#Creating DataFrame
df=pd.DataFrame()

temp=re.findall(Response,data)
df["Response"]=pd.DataFrame(temp)

temp=re.findall(Subcase,data)
df["Subcase"]=pd.DataFrame(temp)

temp=re.findall(Frequency,data)
df["Frequency"]=pd.DataFrame(temp)

temp=re.findall(Values,data)
df[['X','Y','Z']]=pd.DataFrame(temp)
```

In [6]:
```
#String into float
df[["Subcase","Frequency"]]=df[["Subcase","Frequency"]].astype("float64")
df[["X","Y","Z"]]=df[["X","Y","Z"]].astype("float64")
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10800 entries, 0 to 10799
Data columns (total 6 columns):
 #   Column     Non-Null Count  Dtype  
---  ------     --------------  -----  
 0   Response   10800 non-null  object 
 1   Subcase    10800 non-null  float64
 2   Frequency  10800 non-null  float64
 3   X          10800 non-null  float64
 4   Y          10800 non-null  float64
 5   Z          10800 non-null  float64
dtypes: float64(5), object(1)
memory usage: 506.4+ KB
```

In [7]:
```
#DataFrame
df
```

Out[7]:

	Response	Subcase	Frequency	X	Y	Z
0	$DISPLACEMENTS	1.0	1.0	0.000145	0.000311	0.000167
1	$DISPLACEMENTS	1.0	2.0	0.000145	0.000311	0.000167
2	$DISPLACEMENTS	1.0	3.0	0.000145	0.000311	0.000167
3	$DISPLACEMENTS	1.0	4.0	0.000145	0.000311	0.000167
4	$DISPLACEMENTS	1.0	5.0	0.000145	0.000311	0.000167
...	...	...	...	...	...	...
10795	$ACCELERATION	12.0	296.0	750.947600	679.311400	522.432500
10796	$ACCELERATION	12.0	297.0	757.839800	652.520600	517.990800
10797	$ACCELERATION	12.0	298.0	765.432200	627.028700	520.440300
10798	$ACCELERATION	12.0	299.0	773.719100	602.734700	529.337700
10799	$ACCELERATION	12.0	300.0	782.697700	579.547000	544.099400

10800 rows × 6 columns

In [8]:
```
#Function to extract data from either X, Y or Z based on subcase number
def fun(x):
    if x["Subcase"]%3==1:
        return x["X"]
    elif x["Subcase"]%3==2:
        return x["Y"]
    else:
        return x["Z"]
```

In [9]:
```
#Function call
df["result"]=df.apply(fun,axis=1)
df
```

Out[9]:

	Response	Subcase	Frequency	X	Y	Z	result
0	$DISPLACEMENTS	1.0	1.0	0.000145	0.000311	0.000167	0.000145
1	$DISPLACEMENTS	1.0	2.0	0.000145	0.000311	0.000167	0.000145
2	$DISPLACEMENTS	1.0	3.0	0.000145	0.000311	0.000167	0.000145
3	$DISPLACEMENTS	1.0	4.0	0.000145	0.000311	0.000167	0.000145
4	$DISPLACEMENTS	1.0	5.0	0.000145	0.000311	0.000167	0.000145
...	...	...	...	...	...	...	...
10795	$ACCELERATION	12.0	296.0	750.947600	679.311400	522.432500	522.432500
10796	$ACCELERATION	12.0	297.0	757.839800	652.520600	517.990800	517.990800
10797	$ACCELERATION	12.0	298.0	765.432200	627.028700	520.440300	520.440300
10798	$ACCELERATION	12.0	299.0	773.719100	602.734700	529.337700	529.337700
10799	$ACCELERATION	12.0	300.0	782.697700	579.547000	544.099400	544.099400

10800 rows × 7 columns

In [10]:
```
#Pivit table
df_result=df.pivot(columns=["Response","Subcase"],index=["Frequency"],values='
df_result
```

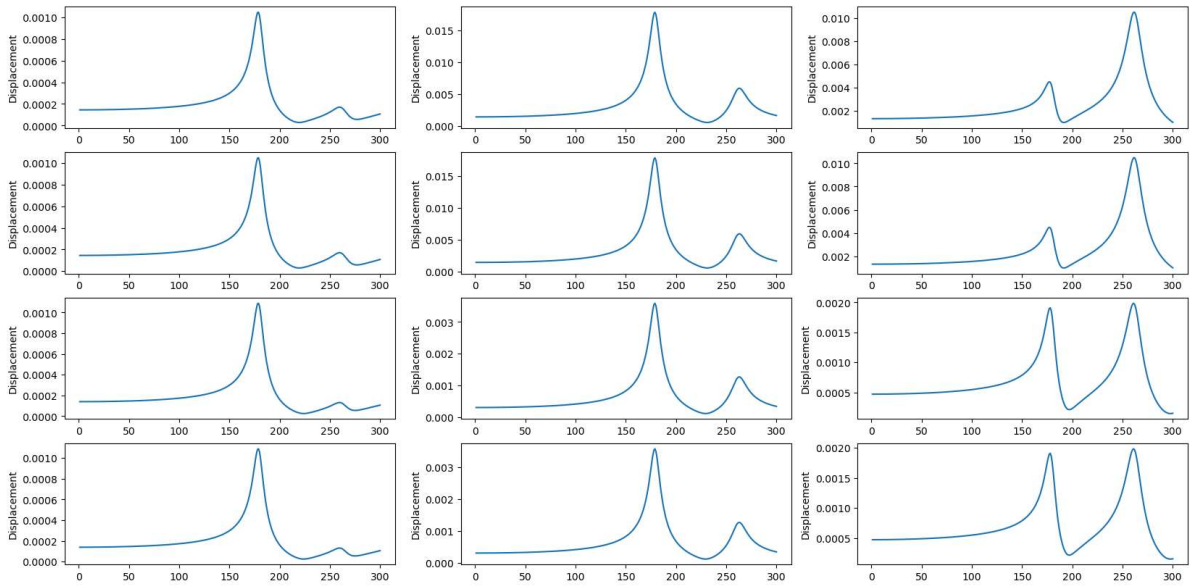
Out[10]:

Response	**$DISPLACEMENTS**								
Subcase	**1.0**	**2.0**	**3.0**	**4.0**	**5.0**	**6.0**	**7.0**	**8.0**	**9.0**
Frequency									
1.0	0.000145	0.001425	0.001316	0.000145	0.001425	0.001316	0.000139	0.000300	0.000
2.0	0.000145	0.001425	0.001316	0.000145	0.001425	0.001316	0.000139	0.000300	0.000
3.0	0.000145	0.001425	0.001316	0.000145	0.001425	0.001316	0.000139	0.000300	0.000
4.0	0.000145	0.001425	0.001316	0.000145	0.001425	0.001316	0.000139	0.000300	0.000
5.0	0.000145	0.001425	0.001317	0.000145	0.001425	0.001317	0.000139	0.000300	0.000
...	...	...	...	...	...	...	...	...	
296.0	0.000097	0.001797	0.001281	0.000097	0.001797	0.001281	0.000096	0.000369	0.000
297.0	0.000099	0.001756	0.001207	0.000099	0.001756	0.001207	0.000099	0.000360	0.000
298.0	0.000102	0.001716	0.001136	0.000102	0.001716	0.001136	0.000101	0.000352	0.000
299.0	0.000105	0.001679	0.001069	0.000105	0.001679	0.001069	0.000104	0.000344	0.000
300.0	0.000107	0.001643	0.001006	0.000107	0.001643	0.001006	0.000106	0.000336	0.000

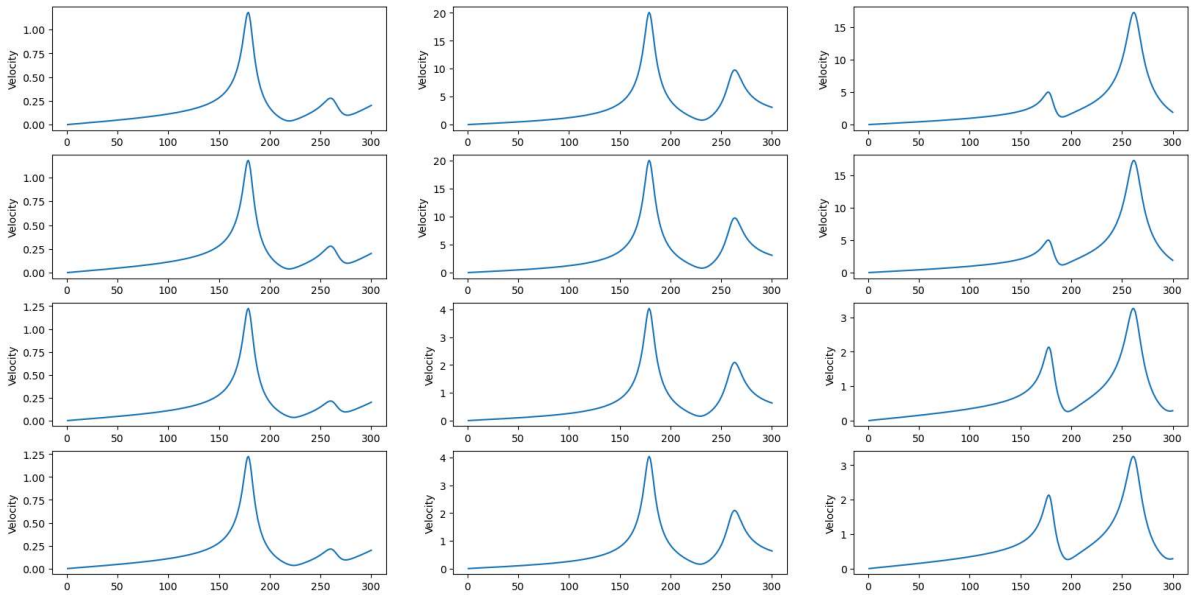
300 rows × 36 columns

In [11]:
```
#Segregating different types of data
Displacement=df_result["$DISPLACEMENTS"]
Velocity=df_result["$VELOCITY"]
Acceleration=df_result["$ACCELERATION"]
```

In [12]:
```
#Visualization: Displacement
plt.figure(figsize=(20,10))
for i in range(1,13):
    plt.subplot(4,3,i)
    plt.plot(Displacement[i])
    plt.ylabel("Displacement")
```



In [13]:
```
#Visualization: Velocity
plt.figure(figsize=(20,10))
for i in range(1,13):
    plt.subplot(4,3,i)
    plt.plot(Velocity[i])
    plt.ylabel("Velocity")
```



In [14]:
```
#Visualization: Acceleration
plt.figure(figsize=(20,10))
for i in range(1,13):
    plt.subplot(4,3,i)
    plt.plot(Acceleration[i],color="b")
    plt.ylabel("Acceleration")
```