```python
import matplotlib.pyplot as plt
from matplotlib import gridspec
import numpy as np
import pandas as pd

import copy

import astropy.units as u
from astropy.constants import k_B


# ================================================================================
# Reading in
# ================================================================================
def read_in(target):
    # data = pd.read_csv("group02_data/processed_data/" + target)
    data = pd.read_csv("processed_data/" + target)
    data.rename(columns={'# x': 'x'}, inplace=True)

    data.loc[:, 'x'] = (np.array(data.loc[:, 'x']) * xpix).to(u.mm)
    data.loc[:, 'y'] = (np.array(data.loc[:, 'y']) * ypix).to(u.mm)
    return data


# ================================================================================
# Analysis
# ================================================================================
def micromotion_sub(variable, data, dt, region):
    datav = data[variable]
```

```
30      datav_four = np.fft.fft(datav)
31      freq = np.fft.fftfreq(len(datav), dt)
32      dffreq = pd.DataFrame({'freq': freq, 'data': datav_four})
33      dffreq_sub = copy.deepcopy(dffreq)
34      dffreq_sub.loc[(np.abs(dffreq_sub['freq']) > region[0]) & (np.abs(dffreq_sub['freq']) <
    region[1]), 'data'] = 0
35
36      datav_sub = np.fft.ifft(dffreq_sub['data']).real
37      return datav, dffreq, datav_sub, dffreq_sub
38
39
40  def vel_calc(data, dt, bins):
41      vel = (data[1:] - data[:-1]) / dt
42      vel = np.histogram(np.abs(vel), bins=bins)
43      df = pd.DataFrame(
44          {'bin_right': vel[1][1:], 'bin_right_squr': vel[1][1:] ** 2, 'count': vel[0], 'err'
    : np.sqrt(vel[0])})
45      return df[df['count'] >= 10]
46
47
48  def calc_mass(fit):
49      t = 298.15 * u.Kelvin
50      m = (fit * (-2 * k_B * t) / (u.mm / u.s) ** 2).to(u.pg)
51      return m
52
53
54  # ================================================================================
55  # Plotting
56  # ================================================================================
```

```
57 def plot_pos(data_init, data_sub, freq_init, freq_sub, dt, labels, img_path, color=None):
58     if color is None:
59         color = color0
60 
61     # Plot setup
62     X = [(1, 3, (2, 3)), (1, 3, 1)]
63     plt.figure(figsize=(18, 3))
64     # plt.subplots_adjust(bottom=0, left=0, top=0.975, right=1)
65 
66     # Plotting Time Domain
67     plt.subplot(*X[0])
68     t = np.linspace(0, len(data_init) * dt, len(data_init))
69     plt.plot(t, data_init, label=labels['data_label_1'], color=color0[0])
70     plt.plot(t, data_sub, label=labels['data_label_2'], color=color0[1])
71     plt.xlim(t[0], t[-1])
72     plt.xlabel(labels['data_xlabel'], fontsize=labels['label_fontsize'])
73     plt.ylabel(labels['data_ylabel'], fontsize=labels['label_fontsize'])
74     plt.title(labels['data_title'], fontsize=labels['title_fontsize'])
75     plt.legend(fontsize=labels['legend_fontsize'])
76 
77     # Plotting Freq Domain
78     plt.subplot(*X[1])
79     plt.plot(freq_init['freq'], np.abs(freq_init['data']), color=color0[0], label=labels['
   freq_label_1'])
80     plt.plot(freq_sub['freq'], np.abs(freq_sub['data']), color=color0[1], label=labels['
   freq_label_2'])
81     plt.xlim(0., np.max(freq_init['freq']))
82     plt.xlabel(labels['freq_xlabel'], fontsize=labels['label_fontsize'])
83     plt.ylabel(labels['freq_ylabel'], fontsize=labels['label_fontsize'])
```

```
 84      plt.title(labels['freq_title'], fontsize=labels['title_fontsize'])
 85      plt.yscale('log')
 86      plt.legend(fontsize=labels['legend_fontsize'])
 87
 88      # Managing final presentation
 89      plt.tight_layout()
 90      plt.savefig(img_path + labels['save_name'])
 91      plt.show()
 92
 93
 94  def plot_vel(data, dt, labels, img_path, color=None):
 95      if color is None:
 96          color = color0
 97
 98      # Plot setup
 99      X = [(1, 3, (1, 2)), (1, 3, 3)]
100      plt.figure(figsize=(18, 3))
101      plt.subplots_adjust(bottom=0, left=0, top=0.975, right=1)
102
103      # Calculating fit
104      fit, res, _, _, _ = np.polyfit(np.array(data['bin_right_squr']), np.array(np.log(data['
     count'])), 1,
105                                     w =np.log(data['err']), cov=True, full=True)
106      fit, cov = np.polyfit(data['bin_right_squr'], np.log(data['count']), 1, w=np.log(data['
     err']), cov=True)
107      chisq_red = float(res / (len(data['bin_right_squr']) - 2))
108      err_a = np.sqrt(cov[0, 0])
109
110      # Plotting Velocities and fit
```

```
111     plt.subplot(*X[0])
112     x = np.linspace(np.min(data['bin_right_squr']), np.max(data['bin_right_squr']), 1000)
113     plt.errorbar(data['bin_right_squr'], data['count'], yerr=data['err'],
114                  linestyle="None", marker="None", ms=7, ecolor=color[0], mfc=color[1], mew=
    0, label=labels['label_1'])
115     plt.plot(x, np.exp(fit[0] * x + fit[1]), color='black')
116     plt.xlabel(labels['xlabel_1'], fontsize=labels['label_fontsize'])
117     plt.ylabel(labels['ylabel_1'], fontsize=labels['label_fontsize'])
118     plt.title(labels['title_1'], fontsize=labels['title_fontsize'])
119     plt.yscale('log')
120     plt.text(np.max(data['bin_right_squr'])*0.85, np.max(data['count'])*0.55,
121              rf"Line fit: ({fit[0]:.2f})$v^2$ + {fit[1]:.2f}" + "\n" + r"$\chi^2_{red} = $"
     + f"{chisq_red:.4f}",
122              fontsize=labels['text_fontsize'], ha='center', va='center')
123
124     # Plotting Residuals
125     plt.subplot(*X[1])
126     plt.errorbar(data['bin_right_squr'], data['count'] - np.exp(fit[0] * data['
    bin_right_squr'] + fit[1]),
127                  yerr=data['err'],
128                  linestyle="None", marker="None", ms=7, ecolor=color[0], mfc=color[1], mew=
    0, label=labels['label_2'])
129     plt.hlines(0, np.min(data['bin_right_squr']), np.max(data['bin_right_squr']), color='
    black')
130     plt.xlabel(labels['xlabel_2'], fontsize=labels['label_fontsize'])
131     plt.ylabel(labels['ylabel_2'], fontsize=labels['label_fontsize'])
132     plt.title(labels['title_2'], fontsize=labels['title_fontsize'])
133
134     plt.tight_layout()
```

```
135     plt.savefig(img_path + labels['save_name'])
136     plt.show()
137     return (fit[0], err_a), chisq_red
138 
139 
140 # ============================================================================
141 # Settings and Running
142 # ============================================================================
143 def main(sets):
144     data = read_in(sets['target'])
145     data, freq, data_sub, freq_sub = micromotion_sub(sets['target_variable'], data, sets['
    time_step'],
146                                                      sets['region'])
147     plot_pos(data, data_sub, freq, freq_sub, sets['time_step'], sets['pos_labels'], sets['
    img_path'],
148              color=sets['colors'])
149     velocities = vel_calc(data_sub, sets['time_step'], sets['bins'])
150     fit, chi2 = plot_vel(velocities, sets['time_step'], sets['vel_labels'], sets['img_path'
    ], color=sets['colors'])
151     print(f"calculated mass: {calc_mass(fit[0]).to(u.pg):.5f} +/- {abs(calc_mass(fit[1]).to
    (u.pg)):.5f}")
152     print(f"chi^2_(red): {chi2:.3f}")
153 
154 
155 if __name__ == "__main__":
156     color0 = ['#3387ec', '#ec9833']
157 
158     xscale = 66
159     yscale = 67
```

```
160     xpix = u.def_unit('xpix', 1 * u.mm / xscale)
161     ypix = u.def_unit('ypix', 1 * u.mm / yscale)
162
163     pos_labels = {
164         'label_fontsize':   20,
165         'title_fontsize':   25,
166         'legend_fontsize':  15,
167         'text_fontsize':    15,
168
169
170         'data_label_1':     'Initial Data',
171         'data_label_2':     r'Inverse FFT (50-80$\rightarrow$0)',
172         'data_xlabel':      'Time (s)',
173         'data_ylabel':      'Position (mm)',
174         'data_title':       'Y Position in Time Domain - 200 Hz',
175         'freq_label_1':     'FFT',
176         'freq_label_2':     r'FFT (50-80$\rightarrow$0)',
177         'freq_xlabel':      'Freq (Hz)',
178         'freq_ylabel':      'Amplitude',
179         'freq_title':       'Freq Domain - 200 Hz',
180         'save_name':        'data_36_y_pos.pdf',
181     }
182     vel_labels = {
183         'label_fontsize':   20,
184         'title_fontsize':   25,
185         'legend_fontsize':  15,
186         'text_fontsize':    15,
187
188         'label_1':          '',
```

```
189         'xlabel_1':         r'Velocity$^2$ (mm/s)$^2$',
190         'ylabel_1':         'Counts',
191         'title_1':          'Velocity (Y) Squared - 200 Hz',
192         'label_2':          '',
193         'xlabel_2':         r'Residuals of Velocity$^2$ (mm/s)$^2$',
194         'ylabel_2':         'Counts-Fit',
195         'title_2':          'Residuals',
196         'save_name':        'data_36_y_vel.pdf',
197     }
198 
199     settings = {
200         'img_path': './images/',
201         'pos_labels': pos_labels,
202         'vel_labels': vel_labels,
203         'colors': color0,
204 
205         'target': 'data_36.csv',
206         'time_step': 1 / 200,
207         'target_variable': 'y',
208         'region': (50, 80),
209         'bins': 1000,
210     }
211     main(settings)
212 
```