Import relevant packages here.

In [18]:

```
import matplotlib.pyplot as plt
import pandas as pd
```

```
             dv        s         a
0     -0.743240  53.5427  1.242570
1     -0.557230  53.6120  1.777920
2     -0.454769  53.6541  0.544107
3     -0.525396  53.7030 -0.294755
4     -0.601285  53.7592 -0.290961
...         ...      ...       ...
73893  0.067944  16.3166 -0.375556
73894  0.066033  16.3146  0.023560
73895  0.011580  16.3034 -0.502929
73896 -0.173098  16.3123 -0.300304
73897 -0.161517  16.3381  0.724566

[73898 rows x 3 columns]
```

Load the data and verify it is loaded correctly.

- **Print it (head, tail, or specific rows, choose a sensible number of rows).**
- **Compare it to the source file.**

In [83]:

```
data = pd.read_csv('cf_data.csv')

print(data.head(-10))
```

```
             dv        s         a
0     -0.743240  53.5427  1.242570
1     -0.557230  53.6120  1.777920
2     -0.454769  53.6541  0.544107
3     -0.525396  53.7030 -0.294755
4     -0.601285  53.7592 -0.290961
...         ...      ...       ...
73893  0.067944  16.3166 -0.375556
73894  0.066033  16.3146  0.023560
73895  0.011580  16.3034 -0.502929
73896 -0.173098  16.3123 -0.300304
73897 -0.161517  16.3381  0.724566

[73898 rows x 3 columns]
```

In the ensuing, you will use `numpy`.

Let's create a grid for the values to plot. But first create two arrays named `dv` and `s` using `numpy.linspace` that hold the grid values at the relevant indices in their respective dimension of the grid.

Create a grid named `a` with zeros using `numpy.zeros` in to which calculated acceleration values can be stored.

Let the grid span:

- **Speed difference `dv` [m/s]**
  - **From -10 till 10**
  - **With 41 evenly spaced values**
- **Headway `s` [m]**
  - **From 0 till 200**
  - **With 21 evenly spaced values**

In [77]:

```
import numpy as np
dv = np.linspace(-10, 10, 41)
s = np.linspace(0, 200, 21)
a = np.zeros((40, 20))
```

Create from the imported data 3 separate `numpy` arrays for each column `dv`, `s` and `a`. (We do this for speed reasons later.)

- **Make sure to name them differently from the arrays that belong to the grid as above.**
- **You can access the data of each column in a `DataFrame` using `data.xxx` where `xxx` is the column name (not as a string).**
- **Use the method `to_numpy()` to convert a column to a `numpy` array.**

In [86]:

```
DV = data["dv"].to_numpy()
S = data["s"].to_numpy()
A = data["a"].to_numpy()
```

```
[-0.74324  -0.55723  -0.454769 ...   5.13764    5.15348    5.25868 ]
```

Create an algorithm that calculates all the acceleration values and stores them in the grid. The algorithm is described visually in the last part of the lecture. At each grid point, it calculates a weighted mean of all measurements. The weights are given by an exponential function, based on the 'distance' between the grid point, and the measurement values of `dv` and `s`. To get you started, how many `for`-loops do you need?

For this you will need `math`.
Use an *upsilon* of 1.5m/s and a *sigma* of 30m.

Warning: **This calculation may take some time. So:**

- **Print a line for each iteration of the outer-most `for`-loop that shows you the progress.**
- **Test you code by running it only on the first 50 measurements of the data.**

I took quite some time figuring out the following code. However it seems it won't work even though I tried quite a few things

In [95]:

```
# Define constants
upsilon = 1.5  # m/s
sigma = 30.0   # m

# Create an empty grid to store acceleration values
grid_shape = (40, 20)  # Define the shape of the grid
acceleration_grid = np.zeros(grid_shape)  # Initialize with zeros

# Iterate over each grid point
for i in range(grid_shape[0]):  # Outermost loop for rows
    for j in range(grid_shape[1]):  # Second loop for columns
        # Initialize variables to calculate weighted mean
        weighted_sum = 0.0
        total_weight = 0.0

        # Iterate over each measurement (considering only the first 50 for testing)
        for k in range(50):  # Third loop for measurements
            # Calculate the Euclidean distance between the grid point and measurement
            distance = math.sqrt((i - DV[k])**2 + (j - S[k])**2)

            # Calculate the weight using the exponential function
            weight = math.exp(-distance / sigma)
```

```python
            # Calculate the weighted contribution to the sum
            weighted_sum += weight * (A[k] - upsilon)

            # Accumulate the total weight
            total_weight += weight

        # Calculate the weighted mean and store it in the grid
        if total_weight > 0:
            acceleration_grid[i, j] = weighted_sum / total_weight

        # Print progress
        print(f"Progress: ({i}/{grid_shape[0] - 1}, {j}/{grid_shape[1] - 1})")

# Now, the acceleration_grid contains the calculated acceleration values
```

```
Progress: (0/39, 0/19)
Progress: (0/39, 1/19)
Progress: (0/39, 2/19)
Progress: (0/39, 3/19)
Progress: (0/39, 4/19)
Progress: (0/39, 5/19)
Progress: (0/39, 6/19)
Progress: (0/39, 7/19)
Progress: (0/39, 8/19)
Progress: (0/39, 9/19)
Progress: (0/39, 10/19)
Progress: (0/39, 11/19)
Progress: (0/39, 12/19)
Progress: (0/39, 13/19)
Progress: (0/39, 14/19)
Progress: (0/39, 15/19)
Progress: (0/39, 16/19)
Progress: (0/39, 17/19)
Progress: (0/39, 18/19)
Progress: (0/39, 19/19)
Progress: (1/39, 0/19)
Progress: (1/39, 1/19)
Progress: (1/39, 2/19)
Progress: (1/39, 3/19)
Progress: (1/39, 4/19)
Progress: (1/39, 5/19)
Progress: (1/39, 6/19)
Progress: (1/39, 7/19)
Progress: (1/39, 8/19)
Progress: (1/39, 9/19)
Progress: (1/39, 10/19)
Progress: (1/39, 11/19)
Progress: (1/39, 12/19)
Progress: (1/39, 13/19)
Progress: (1/39, 14/19)
Progress: (1/39, 15/19)
Progress: (1/39, 16/19)
Progress: (1/39, 17/19)
Progress: (1/39, 18/19)
Progress: (1/39, 19/19)
Progress: (2/39, 0/19)
Progress: (2/39, 1/19)
Progress: (2/39, 2/19)
Progress: (2/39, 3/19)
Progress: (2/39, 4/19)
Progress: (2/39, 5/19)
Progress: (2/39, 6/19)
Progress: (2/39, 7/19)
Progress: (2/39, 8/19)
Progress: (2/39, 9/19)
Progress: (2/39, 10/19)
Progress: (2/39, 11/19)
Progress: (2/39, 12/19)
Progress: (2/39, 13/19)
Progress: (2/39, 14/19)
Progress: (2/39, 15/19)
```

```
Progress: (2/39, 16/19)
Progress: (2/39, 17/19)
Progress: (2/39, 18/19)
Progress: (2/39, 19/19)
Progress: (3/39, 0/19)
Progress: (3/39, 1/19)
Progress: (3/39, 2/19)
Progress: (3/39, 3/19)
Progress: (3/39, 4/19)
Progress: (3/39, 5/19)
Progress: (3/39, 6/19)
Progress: (3/39, 7/19)
Progress: (3/39, 8/19)
Progress: (3/39, 9/19)
Progress: (3/39, 10/19)
Progress: (3/39, 11/19)
Progress: (3/39, 12/19)
Progress: (3/39, 13/19)
Progress: (3/39, 14/19)
Progress: (3/39, 15/19)
Progress: (3/39, 16/19)
Progress: (3/39, 17/19)
Progress: (3/39, 18/19)
Progress: (3/39, 19/19)
Progress: (4/39, 0/19)
Progress: (4/39, 1/19)
Progress: (4/39, 2/19)
Progress: (4/39, 3/19)
Progress: (4/39, 4/19)
Progress: (4/39, 5/19)
Progress: (4/39, 6/19)
Progress: (4/39, 7/19)
Progress: (4/39, 8/19)
Progress: (4/39, 9/19)
Progress: (4/39, 10/19)
Progress: (4/39, 11/19)
Progress: (4/39, 12/19)
Progress: (4/39, 13/19)
Progress: (4/39, 14/19)
Progress: (4/39, 15/19)
Progress: (4/39, 16/19)
Progress: (4/39, 17/19)
Progress: (4/39, 18/19)
Progress: (4/39, 19/19)
Progress: (5/39, 0/19)
Progress: (5/39, 1/19)
Progress: (5/39, 2/19)
Progress: (5/39, 3/19)
Progress: (5/39, 4/19)
Progress: (5/39, 5/19)
Progress: (5/39, 6/19)
Progress: (5/39, 7/19)
Progress: (5/39, 8/19)
Progress: (5/39, 9/19)
Progress: (5/39, 10/19)
Progress: (5/39, 11/19)
Progress: (5/39, 12/19)
Progress: (5/39, 13/19)
Progress: (5/39, 14/19)
Progress: (5/39, 15/19)
Progress: (5/39, 16/19)
Progress: (5/39, 17/19)
Progress: (5/39, 18/19)
Progress: (5/39, 19/19)
Progress: (6/39, 0/19)
Progress: (6/39, 1/19)
Progress: (6/39, 2/19)
Progress: (6/39, 3/19)
Progress: (6/39, 4/19)
Progress: (6/39, 5/19)
Progress: (6/39, 6/19)
Progress: (6/39, 7/19)
```

```
Progress: (6/39, 8/19)
Progress: (6/39, 9/19)
Progress: (6/39, 10/19)
Progress: (6/39, 11/19)
Progress: (6/39, 12/19)
Progress: (6/39, 13/19)
Progress: (6/39, 14/19)
Progress: (6/39, 15/19)
Progress: (6/39, 16/19)
Progress: (6/39, 17/19)
Progress: (6/39, 18/19)
Progress: (6/39, 19/19)
Progress: (7/39, 0/19)
Progress: (7/39, 1/19)
Progress: (7/39, 2/19)
Progress: (7/39, 3/19)
Progress: (7/39, 4/19)
Progress: (7/39, 5/19)
Progress: (7/39, 6/19)
Progress: (7/39, 7/19)
Progress: (7/39, 8/19)
Progress: (7/39, 9/19)
Progress: (7/39, 10/19)
Progress: (7/39, 11/19)
Progress: (7/39, 12/19)
Progress: (7/39, 13/19)
Progress: (7/39, 14/19)
Progress: (7/39, 15/19)
Progress: (7/39, 16/19)
Progress: (7/39, 17/19)
Progress: (7/39, 18/19)
Progress: (7/39, 19/19)
Progress: (8/39, 0/19)
Progress: (8/39, 1/19)
Progress: (8/39, 2/19)
Progress: (8/39, 3/19)
Progress: (8/39, 4/19)
Progress: (8/39, 5/19)
Progress: (8/39, 6/19)
Progress: (8/39, 7/19)
Progress: (8/39, 8/19)
Progress: (8/39, 9/19)
Progress: (8/39, 10/19)
Progress: (8/39, 11/19)
Progress: (8/39, 12/19)
Progress: (8/39, 13/19)
Progress: (8/39, 14/19)
Progress: (8/39, 15/19)
Progress: (8/39, 16/19)
Progress: (8/39, 17/19)
Progress: (8/39, 18/19)
Progress: (8/39, 19/19)
Progress: (9/39, 0/19)
Progress: (9/39, 1/19)
Progress: (9/39, 2/19)
Progress: (9/39, 3/19)
Progress: (9/39, 4/19)
Progress: (9/39, 5/19)
Progress: (9/39, 6/19)
Progress: (9/39, 7/19)
Progress: (9/39, 8/19)
Progress: (9/39, 9/19)
Progress: (9/39, 10/19)
Progress: (9/39, 11/19)
Progress: (9/39, 12/19)
Progress: (9/39, 13/19)
Progress: (9/39, 14/19)
Progress: (9/39, 15/19)
Progress: (9/39, 16/19)
Progress: (9/39, 17/19)
Progress: (9/39, 18/19)
Progress: (9/39, 19/19)
```

```
Progress: (10/39, 0/19)
Progress: (10/39, 1/19)
Progress: (10/39, 2/19)
Progress: (10/39, 3/19)
Progress: (10/39, 4/19)
Progress: (10/39, 5/19)
Progress: (10/39, 6/19)
Progress: (10/39, 7/19)
Progress: (10/39, 8/19)
Progress: (10/39, 9/19)
Progress: (10/39, 10/19)
Progress: (10/39, 11/19)
Progress: (10/39, 12/19)
Progress: (10/39, 13/19)
Progress: (10/39, 14/19)
Progress: (10/39, 15/19)
Progress: (10/39, 16/19)
Progress: (10/39, 17/19)
Progress: (10/39, 18/19)
Progress: (10/39, 19/19)
Progress: (11/39, 0/19)
Progress: (11/39, 1/19)
Progress: (11/39, 2/19)
Progress: (11/39, 3/19)
Progress: (11/39, 4/19)
Progress: (11/39, 5/19)
Progress: (11/39, 6/19)
Progress: (11/39, 7/19)
Progress: (11/39, 8/19)
Progress: (11/39, 9/19)
Progress: (11/39, 10/19)
Progress: (11/39, 11/19)
Progress: (11/39, 12/19)
Progress: (11/39, 13/19)
Progress: (11/39, 14/19)
Progress: (11/39, 15/19)
Progress: (11/39, 16/19)
Progress: (11/39, 17/19)
Progress: (11/39, 18/19)
Progress: (11/39, 19/19)
Progress: (12/39, 0/19)
Progress: (12/39, 1/19)
Progress: (12/39, 2/19)
Progress: (12/39, 3/19)
Progress: (12/39, 4/19)
Progress: (12/39, 5/19)
Progress: (12/39, 6/19)
Progress: (12/39, 7/19)
Progress: (12/39, 8/19)
Progress: (12/39, 9/19)
Progress: (12/39, 10/19)
Progress: (12/39, 11/19)
Progress: (12/39, 12/19)
Progress: (12/39, 13/19)
Progress: (12/39, 14/19)
Progress: (12/39, 15/19)
Progress: (12/39, 16/19)
Progress: (12/39, 17/19)
Progress: (12/39, 18/19)
Progress: (12/39, 19/19)
Progress: (13/39, 0/19)
Progress: (13/39, 1/19)
Progress: (13/39, 2/19)
Progress: (13/39, 3/19)
Progress: (13/39, 4/19)
Progress: (13/39, 5/19)
Progress: (13/39, 6/19)
Progress: (13/39, 7/19)
Progress: (13/39, 8/19)
Progress: (13/39, 9/19)
Progress: (13/39, 10/19)
Progress: (13/39, 11/19)
```

```
Progress: (13/39, 12/19)
Progress: (13/39, 13/19)
Progress: (13/39, 14/19)
Progress: (13/39, 15/19)
Progress: (13/39, 16/19)
Progress: (13/39, 17/19)
Progress: (13/39, 18/19)
Progress: (13/39, 19/19)
Progress: (14/39, 0/19)
Progress: (14/39, 1/19)
Progress: (14/39, 2/19)
Progress: (14/39, 3/19)
Progress: (14/39, 4/19)
Progress: (14/39, 5/19)
Progress: (14/39, 6/19)
Progress: (14/39, 7/19)
Progress: (14/39, 8/19)
Progress: (14/39, 9/19)
Progress: (14/39, 10/19)
Progress: (14/39, 11/19)
Progress: (14/39, 12/19)
Progress: (14/39, 13/19)
Progress: (14/39, 14/19)
Progress: (14/39, 15/19)
Progress: (14/39, 16/19)
Progress: (14/39, 17/19)
Progress: (14/39, 18/19)
Progress: (14/39, 19/19)
Progress: (15/39, 0/19)
Progress: (15/39, 1/19)
Progress: (15/39, 2/19)
Progress: (15/39, 3/19)
Progress: (15/39, 4/19)
Progress: (15/39, 5/19)
Progress: (15/39, 6/19)
Progress: (15/39, 7/19)
Progress: (15/39, 8/19)
Progress: (15/39, 9/19)
Progress: (15/39, 10/19)
Progress: (15/39, 11/19)
Progress: (15/39, 12/19)
Progress: (15/39, 13/19)
Progress: (15/39, 14/19)
Progress: (15/39, 15/19)
Progress: (15/39, 16/19)
Progress: (15/39, 17/19)
Progress: (15/39, 18/19)
Progress: (15/39, 19/19)
Progress: (16/39, 0/19)
Progress: (16/39, 1/19)
Progress: (16/39, 2/19)
Progress: (16/39, 3/19)
Progress: (16/39, 4/19)
Progress: (16/39, 5/19)
Progress: (16/39, 6/19)
Progress: (16/39, 7/19)
Progress: (16/39, 8/19)
Progress: (16/39, 9/19)
Progress: (16/39, 10/19)
Progress: (16/39, 11/19)
Progress: (16/39, 12/19)
Progress: (16/39, 13/19)
Progress: (16/39, 14/19)
Progress: (16/39, 15/19)
Progress: (16/39, 16/19)
Progress: (16/39, 17/19)
Progress: (16/39, 18/19)
Progress: (16/39, 19/19)
Progress: (17/39, 0/19)
Progress: (17/39, 1/19)
Progress: (17/39, 2/19)
Progress: (17/39, 3/19)
```

```
Progress: (17/39, 4/19)
Progress: (17/39, 5/19)
Progress: (17/39, 6/19)
Progress: (17/39, 7/19)
Progress: (17/39, 8/19)
Progress: (17/39, 9/19)
Progress: (17/39, 10/19)
Progress: (17/39, 11/19)
Progress: (17/39, 12/19)
Progress: (17/39, 13/19)
Progress: (17/39, 14/19)
Progress: (17/39, 15/19)
Progress: (17/39, 16/19)
Progress: (17/39, 17/19)
Progress: (17/39, 18/19)
Progress: (17/39, 19/19)
Progress: (18/39, 0/19)
Progress: (18/39, 1/19)
Progress: (18/39, 2/19)
Progress: (18/39, 3/19)
Progress: (18/39, 4/19)
Progress: (18/39, 5/19)
Progress: (18/39, 6/19)
Progress: (18/39, 7/19)
Progress: (18/39, 8/19)
Progress: (18/39, 9/19)
Progress: (18/39, 10/19)
Progress: (18/39, 11/19)
Progress: (18/39, 12/19)
Progress: (18/39, 13/19)
Progress: (18/39, 14/19)
Progress: (18/39, 15/19)
Progress: (18/39, 16/19)
Progress: (18/39, 17/19)
Progress: (18/39, 18/19)
Progress: (18/39, 19/19)
Progress: (19/39, 0/19)
Progress: (19/39, 1/19)
Progress: (19/39, 2/19)
Progress: (19/39, 3/19)
Progress: (19/39, 4/19)
Progress: (19/39, 5/19)
Progress: (19/39, 6/19)
Progress: (19/39, 7/19)
Progress: (19/39, 8/19)
Progress: (19/39, 9/19)
Progress: (19/39, 10/19)
Progress: (19/39, 11/19)
Progress: (19/39, 12/19)
Progress: (19/39, 13/19)
Progress: (19/39, 14/19)
Progress: (19/39, 15/19)
Progress: (19/39, 16/19)
Progress: (19/39, 17/19)
Progress: (19/39, 18/19)
Progress: (19/39, 19/19)
Progress: (20/39, 0/19)
Progress: (20/39, 1/19)
Progress: (20/39, 2/19)
Progress: (20/39, 3/19)
Progress: (20/39, 4/19)
Progress: (20/39, 5/19)
Progress: (20/39, 6/19)
Progress: (20/39, 7/19)
Progress: (20/39, 8/19)
Progress: (20/39, 9/19)
Progress: (20/39, 10/19)
Progress: (20/39, 11/19)
Progress: (20/39, 12/19)
Progress: (20/39, 13/19)
Progress: (20/39, 14/19)
Progress: (20/39, 15/19)
```

```
Progress: (20/39, 16/19)
Progress: (20/39, 17/19)
Progress: (20/39, 18/19)
Progress: (20/39, 19/19)
Progress: (21/39, 0/19)
Progress: (21/39, 1/19)
Progress: (21/39, 2/19)
Progress: (21/39, 3/19)
Progress: (21/39, 4/19)
Progress: (21/39, 5/19)
Progress: (21/39, 6/19)
Progress: (21/39, 7/19)
Progress: (21/39, 8/19)
Progress: (21/39, 9/19)
Progress: (21/39, 10/19)
Progress: (21/39, 11/19)
Progress: (21/39, 12/19)
Progress: (21/39, 13/19)
Progress: (21/39, 14/19)
Progress: (21/39, 15/19)
Progress: (21/39, 16/19)
Progress: (21/39, 17/19)
Progress: (21/39, 18/19)
Progress: (21/39, 19/19)
Progress: (22/39, 0/19)
Progress: (22/39, 1/19)
Progress: (22/39, 2/19)
Progress: (22/39, 3/19)
Progress: (22/39, 4/19)
Progress: (22/39, 5/19)
Progress: (22/39, 6/19)
Progress: (22/39, 7/19)
Progress: (22/39, 8/19)
Progress: (22/39, 9/19)
Progress: (22/39, 10/19)
Progress: (22/39, 11/19)
Progress: (22/39, 12/19)
Progress: (22/39, 13/19)
Progress: (22/39, 14/19)
Progress: (22/39, 15/19)
Progress: (22/39, 16/19)
Progress: (22/39, 17/19)
Progress: (22/39, 18/19)
Progress: (22/39, 19/19)
Progress: (23/39, 0/19)
Progress: (23/39, 1/19)
Progress: (23/39, 2/19)
Progress: (23/39, 3/19)
Progress: (23/39, 4/19)
Progress: (23/39, 5/19)
Progress: (23/39, 6/19)
Progress: (23/39, 7/19)
Progress: (23/39, 8/19)
Progress: (23/39, 9/19)
Progress: (23/39, 10/19)
Progress: (23/39, 11/19)
Progress: (23/39, 12/19)
Progress: (23/39, 13/19)
Progress: (23/39, 14/19)
Progress: (23/39, 15/19)
Progress: (23/39, 16/19)
Progress: (23/39, 17/19)
Progress: (23/39, 18/19)
Progress: (23/39, 19/19)
Progress: (24/39, 0/19)
Progress: (24/39, 1/19)
Progress: (24/39, 2/19)
Progress: (24/39, 3/19)
Progress: (24/39, 4/19)
Progress: (24/39, 5/19)
Progress: (24/39, 6/19)
Progress: (24/39, 7/19)
```

```
Progress: (24/39, 8/19)
Progress: (24/39, 9/19)
Progress: (24/39, 10/19)
Progress: (24/39, 11/19)
Progress: (24/39, 12/19)
Progress: (24/39, 13/19)
Progress: (24/39, 14/19)
Progress: (24/39, 15/19)
Progress: (24/39, 16/19)
Progress: (24/39, 17/19)
Progress: (24/39, 18/19)
Progress: (24/39, 19/19)
Progress: (25/39, 0/19)
Progress: (25/39, 1/19)
Progress: (25/39, 2/19)
Progress: (25/39, 3/19)
Progress: (25/39, 4/19)
Progress: (25/39, 5/19)
Progress: (25/39, 6/19)
Progress: (25/39, 7/19)
Progress: (25/39, 8/19)
Progress: (25/39, 9/19)
Progress: (25/39, 10/19)
Progress: (25/39, 11/19)
Progress: (25/39, 12/19)
Progress: (25/39, 13/19)
Progress: (25/39, 14/19)
Progress: (25/39, 15/19)
Progress: (25/39, 16/19)
Progress: (25/39, 17/19)
Progress: (25/39, 18/19)
Progress: (25/39, 19/19)
Progress: (26/39, 0/19)
Progress: (26/39, 1/19)
Progress: (26/39, 2/19)
Progress: (26/39, 3/19)
Progress: (26/39, 4/19)
Progress: (26/39, 5/19)
Progress: (26/39, 6/19)
Progress: (26/39, 7/19)
Progress: (26/39, 8/19)
Progress: (26/39, 9/19)
Progress: (26/39, 10/19)
Progress: (26/39, 11/19)
Progress: (26/39, 12/19)
Progress: (26/39, 13/19)
Progress: (26/39, 14/19)
Progress: (26/39, 15/19)
Progress: (26/39, 16/19)
Progress: (26/39, 17/19)
Progress: (26/39, 18/19)
Progress: (26/39, 19/19)
Progress: (27/39, 0/19)
Progress: (27/39, 1/19)
Progress: (27/39, 2/19)
Progress: (27/39, 3/19)
Progress: (27/39, 4/19)
Progress: (27/39, 5/19)
Progress: (27/39, 6/19)
Progress: (27/39, 7/19)
Progress: (27/39, 8/19)
Progress: (27/39, 9/19)
Progress: (27/39, 10/19)
Progress: (27/39, 11/19)
Progress: (27/39, 12/19)
Progress: (27/39, 13/19)
Progress: (27/39, 14/19)
Progress: (27/39, 15/19)
Progress: (27/39, 16/19)
Progress: (27/39, 17/19)
Progress: (27/39, 18/19)
Progress: (27/39, 19/19)
```

```
Progress: (28/39, 0/19)
Progress: (28/39, 1/19)
Progress: (28/39, 2/19)
Progress: (28/39, 3/19)
Progress: (28/39, 4/19)
Progress: (28/39, 5/19)
Progress: (28/39, 6/19)
Progress: (28/39, 7/19)
Progress: (28/39, 8/19)
Progress: (28/39, 9/19)
Progress: (28/39, 10/19)
Progress: (28/39, 11/19)
Progress: (28/39, 12/19)
Progress: (28/39, 13/19)
Progress: (28/39, 14/19)
Progress: (28/39, 15/19)
Progress: (28/39, 16/19)
Progress: (28/39, 17/19)
Progress: (28/39, 18/19)
Progress: (28/39, 19/19)
Progress: (29/39, 0/19)
Progress: (29/39, 1/19)
Progress: (29/39, 2/19)
Progress: (29/39, 3/19)
Progress: (29/39, 4/19)
Progress: (29/39, 5/19)
Progress: (29/39, 6/19)
Progress: (29/39, 7/19)
Progress: (29/39, 8/19)
Progress: (29/39, 9/19)
Progress: (29/39, 10/19)
Progress: (29/39, 11/19)
Progress: (29/39, 12/19)
Progress: (29/39, 13/19)
Progress: (29/39, 14/19)
Progress: (29/39, 15/19)
Progress: (29/39, 16/19)
Progress: (29/39, 17/19)
Progress: (29/39, 18/19)
Progress: (29/39, 19/19)
Progress: (30/39, 0/19)
Progress: (30/39, 1/19)
Progress: (30/39, 2/19)
Progress: (30/39, 3/19)
Progress: (30/39, 4/19)
Progress: (30/39, 5/19)
Progress: (30/39, 6/19)
Progress: (30/39, 7/19)
Progress: (30/39, 8/19)
Progress: (30/39, 9/19)
Progress: (30/39, 10/19)
Progress: (30/39, 11/19)
Progress: (30/39, 12/19)
Progress: (30/39, 13/19)
Progress: (30/39, 14/19)
Progress: (30/39, 15/19)
Progress: (30/39, 16/19)
Progress: (30/39, 17/19)
Progress: (30/39, 18/19)
Progress: (30/39, 19/19)
Progress: (31/39, 0/19)
Progress: (31/39, 1/19)
Progress: (31/39, 2/19)
Progress: (31/39, 3/19)
Progress: (31/39, 4/19)
Progress: (31/39, 5/19)
Progress: (31/39, 6/19)
Progress: (31/39, 7/19)
Progress: (31/39, 8/19)
Progress: (31/39, 9/19)
Progress: (31/39, 10/19)
Progress: (31/39, 11/19)
```

```
Progress: (31/39, 12/19)
Progress: (31/39, 13/19)
Progress: (31/39, 14/19)
Progress: (31/39, 15/19)
Progress: (31/39, 16/19)
Progress: (31/39, 17/19)
Progress: (31/39, 18/19)
Progress: (31/39, 19/19)
Progress: (32/39, 0/19)
Progress: (32/39, 1/19)
Progress: (32/39, 2/19)
Progress: (32/39, 3/19)
Progress: (32/39, 4/19)
Progress: (32/39, 5/19)
Progress: (32/39, 6/19)
Progress: (32/39, 7/19)
Progress: (32/39, 8/19)
Progress: (32/39, 9/19)
Progress: (32/39, 10/19)
Progress: (32/39, 11/19)
Progress: (32/39, 12/19)
Progress: (32/39, 13/19)
Progress: (32/39, 14/19)
Progress: (32/39, 15/19)
Progress: (32/39, 16/19)
Progress: (32/39, 17/19)
Progress: (32/39, 18/19)
Progress: (32/39, 19/19)
Progress: (33/39, 0/19)
Progress: (33/39, 1/19)
Progress: (33/39, 2/19)
Progress: (33/39, 3/19)
Progress: (33/39, 4/19)
Progress: (33/39, 5/19)
Progress: (33/39, 6/19)
Progress: (33/39, 7/19)
Progress: (33/39, 8/19)
Progress: (33/39, 9/19)
Progress: (33/39, 10/19)
Progress: (33/39, 11/19)
Progress: (33/39, 12/19)
Progress: (33/39, 13/19)
Progress: (33/39, 14/19)
Progress: (33/39, 15/19)
Progress: (33/39, 16/19)
Progress: (33/39, 17/19)
Progress: (33/39, 18/19)
Progress: (33/39, 19/19)
Progress: (34/39, 0/19)
Progress: (34/39, 1/19)
Progress: (34/39, 2/19)
Progress: (34/39, 3/19)
Progress: (34/39, 4/19)
Progress: (34/39, 5/19)
Progress: (34/39, 6/19)
Progress: (34/39, 7/19)
Progress: (34/39, 8/19)
Progress: (34/39, 9/19)
Progress: (34/39, 10/19)
Progress: (34/39, 11/19)
Progress: (34/39, 12/19)
Progress: (34/39, 13/19)
Progress: (34/39, 14/19)
Progress: (34/39, 15/19)
Progress: (34/39, 16/19)
Progress: (34/39, 17/19)
Progress: (34/39, 18/19)
Progress: (34/39, 19/19)
Progress: (35/39, 0/19)
Progress: (35/39, 1/19)
Progress: (35/39, 2/19)
Progress: (35/39, 3/19)
```

```
Progress: (35/39, 4/19)
Progress: (35/39, 5/19)
Progress: (35/39, 6/19)
Progress: (35/39, 7/19)
Progress: (35/39, 8/19)
Progress: (35/39, 9/19)
Progress: (35/39, 10/19)
Progress: (35/39, 11/19)
Progress: (35/39, 12/19)
Progress: (35/39, 13/19)
Progress: (35/39, 14/19)
Progress: (35/39, 15/19)
Progress: (35/39, 16/19)
Progress: (35/39, 17/19)
Progress: (35/39, 18/19)
Progress: (35/39, 19/19)
Progress: (36/39, 0/19)
Progress: (36/39, 1/19)
Progress: (36/39, 2/19)
Progress: (36/39, 3/19)
Progress: (36/39, 4/19)
Progress: (36/39, 5/19)
Progress: (36/39, 6/19)
Progress: (36/39, 7/19)
Progress: (36/39, 8/19)
Progress: (36/39, 9/19)
Progress: (36/39, 10/19)
Progress: (36/39, 11/19)
Progress: (36/39, 12/19)
Progress: (36/39, 13/19)
Progress: (36/39, 14/19)
Progress: (36/39, 15/19)
Progress: (36/39, 16/19)
Progress: (36/39, 17/19)
Progress: (36/39, 18/19)
Progress: (36/39, 19/19)
Progress: (37/39, 0/19)
Progress: (37/39, 1/19)
Progress: (37/39, 2/19)
Progress: (37/39, 3/19)
Progress: (37/39, 4/19)
Progress: (37/39, 5/19)
Progress: (37/39, 6/19)
Progress: (37/39, 7/19)
Progress: (37/39, 8/19)
Progress: (37/39, 9/19)
Progress: (37/39, 10/19)
Progress: (37/39, 11/19)
Progress: (37/39, 12/19)
Progress: (37/39, 13/19)
Progress: (37/39, 14/19)
Progress: (37/39, 15/19)
Progress: (37/39, 16/19)
Progress: (37/39, 17/19)
Progress: (37/39, 18/19)
Progress: (37/39, 19/19)
Progress: (38/39, 0/19)
Progress: (38/39, 1/19)
Progress: (38/39, 2/19)
Progress: (38/39, 3/19)
Progress: (38/39, 4/19)
Progress: (38/39, 5/19)
Progress: (38/39, 6/19)
Progress: (38/39, 7/19)
Progress: (38/39, 8/19)
Progress: (38/39, 9/19)
Progress: (38/39, 10/19)
Progress: (38/39, 11/19)
Progress: (38/39, 12/19)
Progress: (38/39, 13/19)
Progress: (38/39, 14/19)
Progress: (38/39, 15/19)
```

```
Progress: (38/39, 16/19)
Progress: (38/39, 17/19)
Progress: (38/39, 18/19)
Progress: (38/39, 19/19)
Progress: (39/39, 0/19)
Progress: (39/39, 1/19)
Progress: (39/39, 2/19)
Progress: (39/39, 3/19)
Progress: (39/39, 4/19)
Progress: (39/39, 5/19)
Progress: (39/39, 6/19)
Progress: (39/39, 7/19)
Progress: (39/39, 8/19)
Progress: (39/39, 9/19)
Progress: (39/39, 10/19)
Progress: (39/39, 11/19)
Progress: (39/39, 12/19)
Progress: (39/39, 13/19)
Progress: (39/39, 14/19)
Progress: (39/39, 15/19)
Progress: (39/39, 16/19)
Progress: (39/39, 17/19)
Progress: (39/39, 18/19)
Progress: (39/39, 19/19)
```

The following code will plot the data for you. Does it make sense when considering:

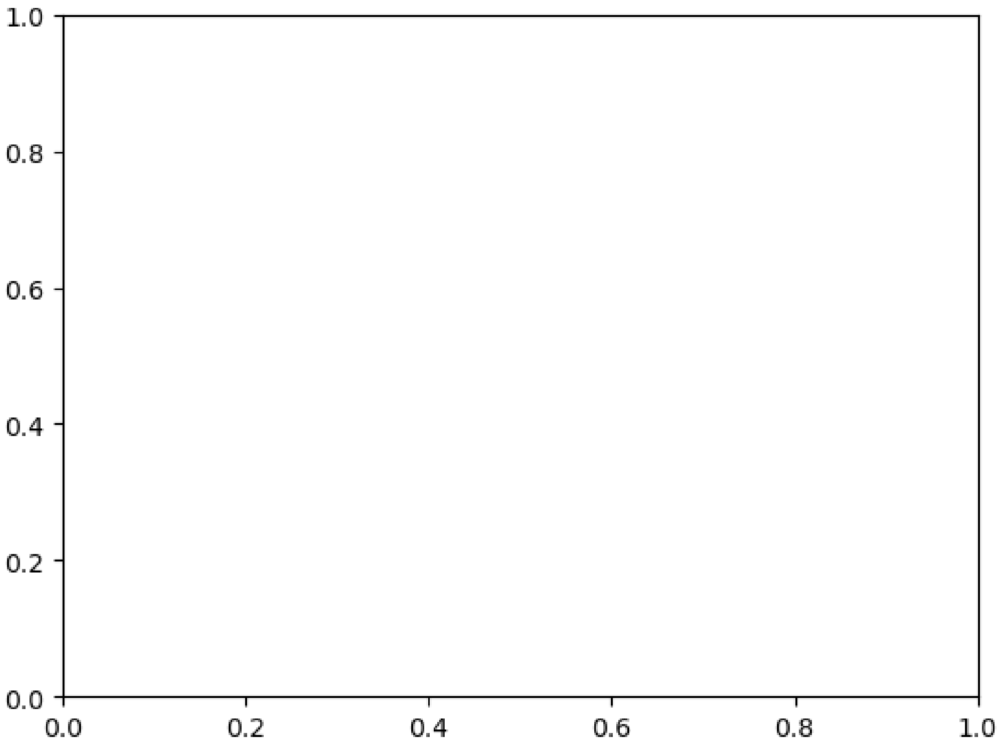
- **Negative (slower than leader) and positive (faster than leader) speed differences?**
- **Small and large headways?**

In [96]:

```
X, Y = np.meshgrid(dv, s)
axs = plt.axes()
p = axs.pcolor(X, Y, a, shading='nearest')
axs.set_title('Acceleration [m/s/s]')
axs.set_xlabel('Speed difference [m/s]')
axs.set_ylabel('Headway [m]')
axs.figure.colorbar(p);
axs.figure.set_size_inches(10, 7)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[96], line 3
      1 X, Y = np.meshgrid(dv, s)
      2 axs = plt.axes()
----> 3 p = axs.pcolor(X, Y, a, shading='nearest')
      4 axs.set_title('Acceleration [m/s/s]')
      5 axs.set_xlabel('Speed difference [m/s]')

File ~\anaconda3\Lib\site-packages\matplotlib\__init__.py:1442, in _preprocess_data.<locals>.inner(ax, data, *args, **kwargs)
   1439 @functools.wraps(func)
   1440 def inner(ax, *args, data=None, **kwargs):
   1441     if data is None:
-> 1442         return func(ax, *map(sanitize_sequence, args), **kwargs)
   1444     bound = new_sig.bind(ax, *args, **kwargs)
   1445     auto_label = (bound.arguments.get(label_namer)
   1446                   or bound.kwargs.get(label_namer))

File ~\anaconda3\Lib\site-packages\matplotlib\axes\_axes.py:5946, in Axes.pcolor(self, shading, alpha, norm, cmap, vmin, vmax, *args, **kwargs)
   5944     shading = mpl.rcParams['pcolor.shading']
   5945 shading = shading.lower()
-> 5946 X, Y, C, shading = self._pcolorargs('pcolor', *args, shading=shading,
   5947                                     kwargs=kwargs)
   5948 Ny, Nx = X.shape
   5950 # convert to MA, if necessary.

File ~\anaconda3\Lib\site-packages\matplotlib\axes\_axes.py:5757, in Axes._pcolorargs(self, funcname, shading, *args, **kwargs)
```

```
   5755 else:    # ['nearest', 'gouraud']:
   5756     if (Nx, Ny) != (ncols, nrows):
-> 5757         raise TypeError('Dimensions of C %s are incompatible with'
   5758                         ' X (%d) and/or Y (%d); see help(%s)' % (
   5759                             C.shape, Nx, Ny, funcname))
   5760     if shading == 'nearest':
   5761         # grid is specified at the center, so define corners
   5762         # at the midpoints between the grid centers and then use the
   5763         # flat algorithm.
   5764         def _interp_grid(X):
   5765             # helper for below

TypeError: Dimensions of C (40, 20) are incompatible with X (1) and/or Y (21); see help(p
color)
```




```
In [ ]:
```