Phys 238 : Entering and Manipulating Data in *Mathematica*

Getting Started

Mathematica remembers definitions, even if you delete them from the notebook. This can sometimes be confusing, so it's often a good idea to simply clear everything and start anew. This is especially true if you use the Evaluation -> Evaluate Notebook menu item.

In[193]:=
```
Clear["Global`*"] (* Clear all variable and function definitions *)
```

Entering Data

Consider the data for static torque from the torsional oscillator experiment. This data consists of {mass, angle} pairs. There are multiple ways to enter the data.

The data is a list of data points. All lists are made of elements in curly braces { } separated by commas. Here, each data point is itself a list, consisting of two numbers (in curly braces) separated by a comma. To see what the data looks like, display it with TableForm[ ].

There are many ways to enter the data:

Typing the data directly:

In[194]:=

```
rawdata = {
  {0, 2.98},
  {50, 2.79},
  {100, 2.59},
  {150, 2.38},
  {200, 2.17},
  {250, 1.96},
  {300, 1.66},
  {350, 1.41},
  {400, 1.21},
  {0, 2.98},
  {-50, 3.20},
  {-100, 3.40},
  {-150, 3.60},
  {-200, 3.81},
  {-250, 4.06},
  {-300, 4.32},
  {-350, 4.56},
  {-400, 4.74},
  {0, 3.00}
 }
```

Out[194]=

```
{{0, 2.98}, {50, 2.79}, {100, 2.59}, {150, 2.38},
 {200, 2.17}, {250, 1.96}, {300, 1.66}, {350, 1.41}, {400, 1.21},
 {0, 2.98}, {-50, 3.2}, {-100, 3.4}, {-150, 3.6}, {-200, 3.81},
 {-250, 4.06}, {-300, 4.32}, {-350, 4.56}, {-400, 4.74}, {0, 3.}}
```

In[195]:=
```
TableForm[rawdata]
```
Out[195]//TableForm=

0	2.98
50	2.79
100	2.59
150	2.38
200	2.17
250	1.96
300	1.66
350	1.41
400	1.21
0	2.98
−50	3.2
−100	3.4
−150	3.6
−200	3.81
−250	4.06
−300	4.32
−350	4.56
−400	4.74
0	3.

Typing the data using the Classroom Assistant: $\begin{pmatrix} \square & \square \\ \square & \square \end{pmatrix}$

From the Palettes -> Classroom Assistant palette, look for the matrix item: $\begin{pmatrix} \square & \square \\ \square & \square \end{pmatrix}$. (Depending on the version of *Mathematica*, it might be in the “Typesetting” menu, or in the “Advanced” menu. (Use Ctrl-Enter to add a new row.)

In[196]:=

$$\text{rawdata} = \begin{pmatrix} 0 & 2.98 \\ 50 & 2.79 \\ 100 & 2.59 \\ 150 & 2.38 \\ 200 & 2.17 \\ 250 & 1.96 \\ 300 & 1.66 \\ 350 & 1.41 \\ 400 & 1.21 \\ 0 & 2.98 \\ -50 & 3.20 \\ -100 & 3.40 \\ -150 & 3.60 \\ -200 & 3.81 \\ -250 & 4.06 \\ -300 & 4.32 \\ -350 & 4.56 \\ -400 & 4.74 \\ 0 & 3.00 \end{pmatrix}$$

Out[196]=

```
{{0, 2.98}, {50, 2.79}, {100, 2.59}, {150, 2.38},
 {200, 2.17}, {250, 1.96}, {300, 1.66}, {350, 1.41}, {400, 1.21},
 {0, 2.98}, {-50, 3.2}, {-100, 3.4}, {-150, 3.6}, {-200, 3.81},
 {-250, 4.06}, {-300, 4.32}, {-350, 4.56}, {-400, 4.74}, {0, 3.}}
```

Reading from a file:

To read the data from a text file in the same directory as your notebook, first tell *Mathematica* what directory to use. I like to keep the data file in the same directory as the notebook.

In[197]:=

```
SetDirectory[NotebookDirectory[]]
```

Out[197]=

```
/Users/doughera/notes/www/public_html/courses/phys238-2026/notes/torsion
```

Typically, files might have header information (names of columns, dates, etc.) There might also be other comments. Sometimes the columns are separated by spaces, tabls, or commas. It is usually helpful to see what the raw data file looks like. The FilePrint[ ] command prints out the contents of the file so you can look at it and assess what to do next.

In[198]:=
```
FilePrint["TO-static-20250212.csv"]
```

```
"Mass (g)", "Raw Angle", "Voltage"
0, 2.98, -0.0456
50, 2.79, -0.4083
100, 2.59, -0.7733
150, 2.38, -1.172
200, 2.17, -1.556
250, 1.96, -1.958
300, 1.66, -2.483
350, 1.41, -2.726
400, 1.21, -2.566
0, 2.98, -0.0606
-50, 3.20, 0.3049
-100, 3.40, 0.6583
-150, 3.60, 1.028
-200, 3.81, 1.387
-250, 4.06, 1.812
-300, 4.32, 2.237
-350, 4.56, 2.502
-400, 4.74, 2.369
0, 3.00, -0.04502
```

This particular file has 3 columns, separated by commas. There is a single header line at the top giving the column titles.

Importing as a CSV (Comma Separated Values) file

If you know the import format, you can specify it explicitly:

In[199]:=
```
rawdata = Import["TO-static-20250212.csv", "CSV"]
```

Out[199]=
```
{{Mass (g), Raw Angle, Voltage}, {0, 2.98, -0.0456}, {50, 2.79, -0.4083},
 {100, 2.59, -0.7733}, {150, 2.38, -1.172}, {200, 2.17, -1.556},
 {250, 1.96, -1.958}, {300, 1.66, -2.483}, {350, 1.41, -2.726},
 {400, 1.21, -2.566}, {0, 2.98, -0.0606}, {-50, 3.2, 0.3049}, {-100, 3.4, 0.6583},
 {-150, 3.6, 1.028}, {-200, 3.81, 1.387}, {-250, 4.06, 1.812},
 {-300, 4.32, 2.237}, {-350, 4.56, 2.502}, {-400, 4.74, 2.369}, {0, 3., -0.04502}}
```

In[200]:=
```
TableForm[rawdata]
```
Out[200]//TableForm=

Mass (g)	Raw Angle	Voltage
0	2.98	-0.0456
50	2.79	-0.4083
100	2.59	-0.7733
150	2.38	-1.172
200	2.17	-1.556
250	1.96	-1.958
300	1.66	-2.483
350	1.41	-2.726
400	1.21	-2.566
0	2.98	-0.0606
-50	3.2	0.3049
-100	3.4	0.6583
-150	3.6	1.028
-200	3.81	1.387
-250	4.06	1.812
-300	4.32	2.237
-350	4.56	2.502
-400	4.74	2.369
0	3.	-0.04502

In[201]:=
```
Dimensions[rawdata] (* It has 20 rows and 3 columns *)
```
Out[201]=
```
{20, 3}
```

Selecting and Calculating With Data

No matter how you import data, you will likely need to make some adjustments or calculations. Three very useful commands are Select[ ], Sort[ ], and Table[ ]. For simplicity, assume you import the data as a CSV file.

In[202]:=
```
rawdata = Import["TO-static-20250212.csv", "CSV"];
TableForm[rawdata]
```

Out[203]//TableForm=

Mass (g)	Raw Angle	Voltage
0	2.98	-0.0456
50	2.79	-0.4083
100	2.59	-0.7733
150	2.38	-1.172
200	2.17	-1.556
250	1.96	-1.958
300	1.66	-2.483
350	1.41	-2.726
400	1.21	-2.566
0	2.98	-0.0606
-50	3.2	0.3049
-100	3.4	0.6583
-150	3.6	1.028
-200	3.81	1.387
-250	4.06	1.812
-300	4.32	2.237
-350	4.56	2.502
-400	4.74	2.369
0	3.	-0.04502

Looking at parts of an array:

The [[ ]] notation is used to look at an element of an array.

In[204]:=
```
rawdata[[2]] (* The second row *)
```

Out[204]=
```
{0, 2.98, -0.0456}
```

To look at the 3rd item on the second row, use the [[row, column]] notation

In[205]:=
```
rawdata[[2, 3]]
```

Out[205]=
```
-0.0456
```

The special name All can also be used:

In[206]:=
```
rawdata[[All, 3]] (* All rows, column 3 *)
```

Out[206]=
```
{Voltage, -0.0456, -0.4083, -0.7733, -1.172, -1.556, -1.958, -2.483, -2.726, -2.566,
 -0.0606, 0.3049, 0.6583, 1.028, 1.387, 1.812, 2.237, 2.502, 2.369, -0.04502}
```

In[207]:=
```
rawdata[[All, {2, 3}]] (* All rows, columns 2 and 3. *)
```
Out[207]=
```
{{Raw Angle, Voltage}, {2.98, -0.0456}, {2.79, -0.4083},
 {2.59, -0.7733}, {2.38, -1.172}, {2.17, -1.556}, {1.96, -1.958},
 {1.66, -2.483}, {1.41, -2.726}, {1.21, -2.566}, {2.98, -0.0606},
 {3.2, 0.3049}, {3.4, 0.6583}, {3.6, 1.028}, {3.81, 1.387}, {4.06, 1.812},
 {4.32, 2.237}, {4.56, 2.502}, {4.74, 2.369}, {3., -0.04502}}
```

In[208]:=
```
rawdata[[2, All]] (* Row 2, all columns *)
```
Out[208]=
```
{0, 2.98, -0.0456}
```

The Select Command:

The basic syntax for the Select command is Select[data, criteria] (see the online help for full details The Select command loops through the data and returns all points for which the criteria are true. Along the way, it sets the symbol '#' equal to each element of the data. Since # is itself a list (with 3 elements) you can look at the first, second, and third elements of each point with #[[1]], #[[2]], and #[[3]]. (You must include the '&' at the end.) You can combine tests with '&&' to mean logical "AND". You would use '||' to mean logical OR. Thus to select lines where the first, second, and third elements are all numbers, you could do

In[209]:=
```
data = Select[rawdata, NumberQ[#[[1]] && NumberQ[#[[2]] && NumberQ[#[[3]] &]
```
Out[209]=
```
{{0, 2.98, -0.0456}, {50, 2.79, -0.4083}, {100, 2.59, -0.7733},
 {150, 2.38, -1.172}, {200, 2.17, -1.556}, {250, 1.96, -1.958}, {300, 1.66, -2.483},
 {350, 1.41, -2.726}, {400, 1.21, -2.566}, {0, 2.98, -0.0606}, {-50, 3.2, 0.3049},
 {-100, 3.4, 0.6583}, {-150, 3.6, 1.028}, {-200, 3.81, 1.387}, {-250, 4.06, 1.812},
 {-300, 4.32, 2.237}, {-350, 4.56, 2.502}, {-400, 4.74, 2.369}, {0, 3., -0.04502}}
```

Equivalently, you can use the VectorQ function to apply a criterion to every element of a list:

In[210]:=
```
data = Select[rawdata, VectorQ[#, NumberQ] &]
```
Out[210]=
```
{{0, 2.98, -0.0456}, {50, 2.79, -0.4083}, {100, 2.59, -0.7733},
 {150, 2.38, -1.172}, {200, 2.17, -1.556}, {250, 1.96, -1.958}, {300, 1.66, -2.483},
 {350, 1.41, -2.726}, {400, 1.21, -2.566}, {0, 2.98, -0.0606}, {-50, 3.2, 0.3049},
 {-100, 3.4, 0.6583}, {-150, 3.6, 1.028}, {-200, 3.81, 1.387}, {-250, 4.06, 1.812},
 {-300, 4.32, 2.237}, {-350, 4.56, 2.502}, {-400, 4.74, 2.369}, {0, 3., -0.04502}}
```

To select only points where the mass is between -200 and +200 (including the endpoints) you could do

In[211]:=
```
somedata = Select[data, -200 ≤ #[[1]] ≤ 200 &]
```
Out[211]=
```
{{0, 2.98, -0.0456}, {50, 2.79, -0.4083}, {100, 2.59, -0.7733},
 {150, 2.38, -1.172}, {200, 2.17, -1.556}, {0, 2.98, -0.0606}, {-50, 3.2, 0.3049},
 {-100, 3.4, 0.6583}, {-150, 3.6, 1.028}, {-200, 3.81, 1.387}, {0, 3., -0.04502}}
```

Alternatively, if you know you simply want to drop the first row, you can use the Drop[ ] command.

In[212]:=
```
Drop[rawdata, 1]
```
Out[212]=
```
{{0, 2.98, -0.0456}, {50, 2.79, -0.4083}, {100, 2.59, -0.7733},
 {150, 2.38, -1.172}, {200, 2.17, -1.556}, {250, 1.96, -1.958}, {300, 1.66, -2.483},
 {350, 1.41, -2.726}, {400, 1.21, -2.566}, {0, 2.98, -0.0606}, {-50, 3.2, 0.3049},
 {-100, 3.4, 0.6583}, {-150, 3.6, 1.028}, {-200, 3.81, 1.387}, {-250, 4.06, 1.812},
 {-300, 4.32, 2.237}, {-350, 4.56, 2.502}, {-400, 4.74, 2.369}, {0, 3., -0.04502}}
```

In[213]:=
```
SortBy[data, #[[2]] &]
```
Out[213]=
```
{{400, 1.21, -2.566}, {350, 1.41, -2.726}, {300, 1.66, -2.483}, {250, 1.96, -1.958},
 {200, 2.17, -1.556}, {150, 2.38, -1.172}, {100, 2.59, -0.7733},
 {50, 2.79, -0.4083}, {0, 2.98, -0.0606}, {0, 2.98, -0.0456}, {0, 3., -0.04502},
 {-50, 3.2, 0.3049}, {-100, 3.4, 0.6583}, {-150, 3.6, 1.028}, {-200, 3.81, 1.387},
 {-250, 4.06, 1.812}, {-300, 4.32, 2.237}, {-350, 4.56, 2.502}, {-400, 4.74, 2.369}}
```

The SortBy Command:

SortBy [*list*, *f*]

The SortBy[ ] command sorts the elements of *list* in the order defined by applying *f* to each of them in term. This is useful for sorting data arrays in a variety of ways. For example, you might want to sort the data by mass values (the first element in the array). (This uses the same # syntax as the Select command.)

In[214]:=
```
sorted = SortBy[data, #[[1]] &]; (* The ';' suppresses
 printing this line so we can make a pretty table instead. *)
TableForm[sorted]
```

Out[215]//TableForm=

−400	4.74	2.369
−350	4.56	2.502
−300	4.32	2.237
−250	4.06	1.812
−200	3.81	1.387
−150	3.6	1.028
−100	3.4	0.6583
−50	3.2	0.3049
0	2.98	−0.0606
0	2.98	−0.0456
0	3.	−0.04502
50	2.79	−0.4083
100	2.59	−0.7733
150	2.38	−1.172
200	2.17	−1.556
250	1.96	−1.958
300	1.66	−2.483
350	1.41	−2.726
400	1.21	−2.566

You can also sort by the first or last elements in each row with the ‘First’ or ‘Last’ operators instead of #[[1]] or #[[-1]].

In[216]:=
```
sorted = SortBy[data, Last];
TableForm[sorted]
```

Out[217]//TableForm=

350	1.41	−2.726
400	1.21	−2.566
300	1.66	−2.483
250	1.96	−1.958
200	2.17	−1.556
150	2.38	−1.172
100	2.59	−0.7733
50	2.79	−0.4083
0	2.98	−0.0606
0	2.98	−0.0456
0	3.	−0.04502
−50	3.2	0.3049
−100	3.4	0.6583
−150	3.6	1.028
−200	3.81	1.387
−250	4.06	1.812
−300	4.32	2.237
−400	4.74	2.369
−350	4.56	2.502

The Table Command:

Suppose you want to create a new table with {angle, mass} pairs reversed from the input, and you also want to convert the masses from gram to kilogram. The Table[ ] function is a handy way to do that.

First, note that the general syntax to generate a counter 'i' that runs from 1 to the length of the data is

In[218]:=
```
Table[i, {i, 1, Length[data]}]
```
Out[218]=
```
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19}
```

To create our desired array, instead of just printing out the counter, we want to create a list (in curly brackets) of {angle, mass} pairs. Since angle is the second column, you use data[[i, 2]] to get angle and [[i, 1]] to get the mass. The '/1000' converts the masses to kilograms.

In[219]:=
```
massvsAngle = Table[ {data[[i, 2]], data[[i, 1]] / 1000. }, {i, 1, Length[data]}];
```

In[220]:=
```
TableForm[massvsAngle] (* Check what it looks like *)
```
Out[220]//TableForm=

2.98	0.
2.79	0.05
2.59	0.1
2.38	0.15
2.17	0.2
1.96	0.25
1.66	0.3
1.41	0.35
1.21	0.4
2.98	0.
3.2	-0.05
3.4	-0.1
3.6	-0.15
3.81	-0.2
4.06	-0.25
4.32	-0.3
4.56	-0.35
4.74	-0.4
3.	0.

When you want to transform or manipulate your data in some way, the Table[ ] function is often a good approach. There often are more succinct ways to write things in Mathematica, but the Table[ ] function is often the easiest to read and understand.