BPhO Computational Challenge

# Seminar 02: An experimental data processing pipeline

Dr Andrew French.

December 2021.

BPhO British Physics Olympiad

# Experimental data processing pipeline using Excel & MATLAB

Raw data in Excel

**Import** into MATLAB. Assign spreadsheet columns to arrays e.g. `x`, `y`…

**Perform analysis**

- Averages
- Compute uncertainty
- Scaling
- Offset removal
- Linearization
- Line of best fit …

**Plot** data + error bars, *underlaid* with model curve

**Plot** data vs model i.e. a $y = x$ graph and Perform $y = mx$ line of best fit

**Plot** linearized graph and use to determine Model parameters from gradient (and intercept if $y = mx + c$, not $y = mx$ fit)

# Example using pendulum data

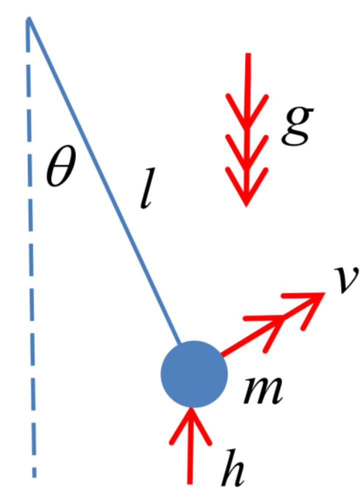
Raw data in Excel

| | length of pendulum L /m | ten periods /s | ten periods /s | ten periods /s |
|---|---|---|---|---|
| 2 | 0.1 | 6.25 | 6.25 | 6.45 |
| 3 | 0.15 | 7.86 | 7.84 | 7.85 |
| 4 | 0.2 | 8.87 | 8.94 | 8.93 |
| 5 | 0.25 | 10.07 | 10.14 | 10.11 |
| 6 | 0.3 | 11.17 | 10.97 | 10.93 |
| 7 | 0.35 | 11.7 | 11.82 | 11.79 |
| 8 | 0.4 | 12.67 | 12.69 | 12.69 |
| 9 | 0.45 | 13.55 | 13.55 | 13.49 |
| 10 | 0.5 | 14.38 | 14.33 | 14.35 |
| 11 | 0.55 | 14.94 | 14.86 | 14.91 |
| 12 | 0.6 | 15.66 | 15.81 | 15.76 |
| 13 | 0.65 | 16 | 15.96 | 16 |
| 14 | 0.7 | 16.9 | 16.92 | 16.92 |
| 15 | 0.75 | 17.26 | 17.47 | 17.35 |
| 16 | 0.8 | 17.95 | 17.97 | 18.11 |
| 17 | 0.85 | 18.67 | 18.72 | 18.65 |
| 18 | 0.9 | 19.19 | 19.23 | 19.17 |
| 19 | 0.95 | 19.89 | 19.85 | 19.82 |
| 20 | 1 | 20.24 | 20.23 | 20.24 |
| 21 | 0.72 | 17.11 | 17.14 | 17.17 |

Pendulum 2020

Data recorded manually or via a datalogger in an Excel spreadsheet.

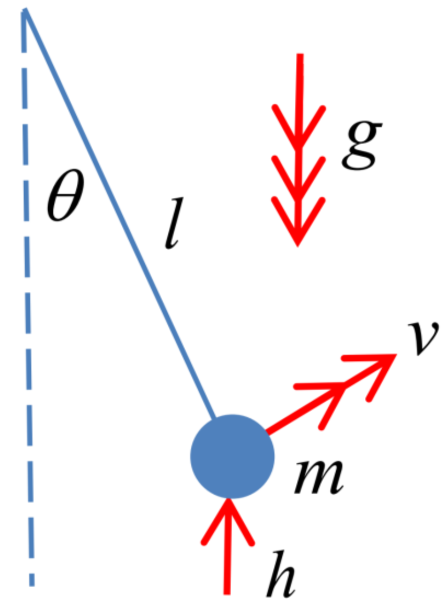
If done manually, I would also recommend a basic graph **plot-as-you-go as a sanity check**, and to guide students to *take more data points* around changes of gradient such as peaks etc.



$$ml\ddot{\theta} = -mg\sin\theta \approx -mg\theta \quad \text{NEWTON II}$$

$$\therefore \ddot{\theta} = -\frac{g}{l}\theta$$

$T$ is the pendulum period




$$\text{SHM:} \quad \ddot{\theta} = -\left(\frac{2\pi}{T}\right)^2\theta = -\omega^2\theta$$

$$\theta(t) = \theta_0\cos\left(\frac{2\pi t}{T}\right) = \theta_0\cos\omega t$$

$$\therefore T = 2\pi\sqrt{\frac{l}{g}} \quad T^2 = 4\pi^2\frac{l}{g} \quad g\underbrace{T^2}_{x} = \underbrace{4\pi^2 l}_{y} \quad \text{i.e.} \quad y = gx$$

**Simple Harmonic Motion** (SHM) of a pendulum

* Ignore air resistance
* Small angle approximation i.e. $\theta \ll 1\,\text{radian}$

**MEASURING g VIA A PENDULUM** 03/06/2020

| length of pendulum L /m | ten periods /s | ten periods /s | ten periods /s | ten periods /s | period T /s | x = T^2 | y = 4*pi^2 * L |
|---|---|---|---|---|---|---|---|
| 0.1 | 6.25 | 6.25 | 6.45 | 6.32 | 0.63 | 0.40 | 3.95 |
| 0.15 | 7.86 | 7.84 | 7.85 | 7.85 | 0.79 | 0.62 | 5.92 |
| 0.2 | 8.87 | 8.94 | 8.93 | 8.91 | 0.89 | 0.79 | 7.90 |
| 0.25 | 10.07 | 10.14 | 10.11 | 10.11 | 1.01 | 1.02 | 9.87 |
| 0.3 | 11.17 | 10.97 | 10.93 | 11.02 | 1.10 | 1.22 | 11.84 |
| 0.35 | 11.7 | 11.82 | 11.79 | 11.77 | 1.18 | 1.39 | 13.82 |
| 0.4 | 12.67 | 12.69 | 12.69 | 12.68 | 1.27 | 1.61 | 15.79 |
| 0.45 | 13.55 | 13.55 | 13.49 | 13.53 | 1.35 | 1.83 | 17.77 |
| 0.5 | 14.38 | 14.33 | 14.35 | 14.35 | 1.44 | 2.06 | 19.74 |
| 0.55 | 14.94 | 14.86 | 14.91 | 14.90 | 1.49 | 2.22 | 21.71 |
| 0.6 | 15.66 | 15.81 | 15.76 | 15.74 | 1.57 | 2.48 | 23.69 |
| 0.65 | 16 | 15.96 | 16 | 15.99 | 1.60 | 2.56 | 25.66 |
| 0.7 | 16.9 | 16.92 | 16.92 | 16.91 | 1.69 | 2.86 | 27.63 |
| 0.75 | 17.26 | 17.47 | 17.35 | 17.36 | 1.74 | 3.01 | 29.61 |
| 0.8 | 17.95 | 17.97 | 18.11 | 18.01 | 1.80 | 3.24 | 31.58 |
| 0.85 | 18.67 | 18.72 | 18.65 | 18.68 | 1.87 | 3.49 | 33.56 |
| 0.9 | 19.19 | 19.23 | 19.17 | 19.20 | 1.92 | 3.69 | 35.53 |
| 0.95 | 19.89 | 19.85 | 19.82 | 19.85 | 1.99 | 3.94 | 37.50 |
| 1 | 20.24 | 20.23 | 20.24 | 20.24 | 2.02 | 4.10 | 39.48 |
| 0.72 | 17.11 | 17.14 | 17.17 | 17.14 | 1.71 | 2.94 | 28.42 |

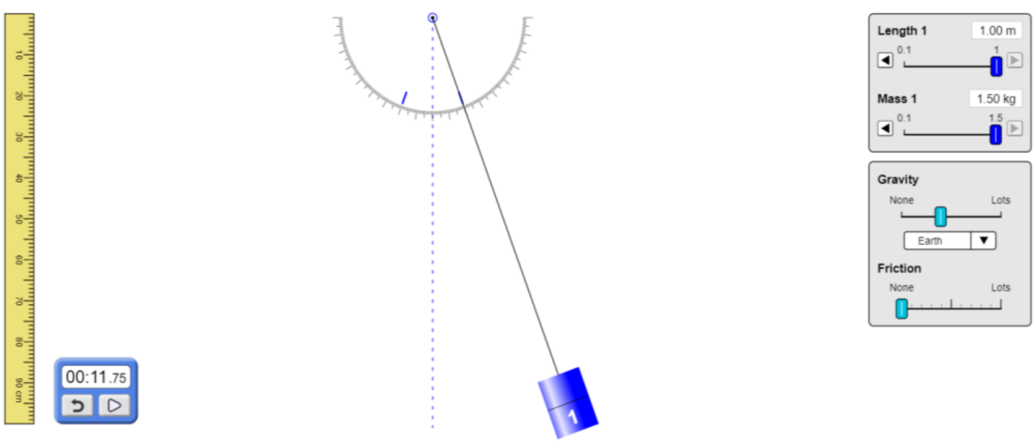
$$ml\ddot{\theta} = -mg\sin\theta \approx -mg\theta \quad \text{NEWTON II}$$

$$\therefore \ddot{\theta} = -\frac{g}{l}\theta$$

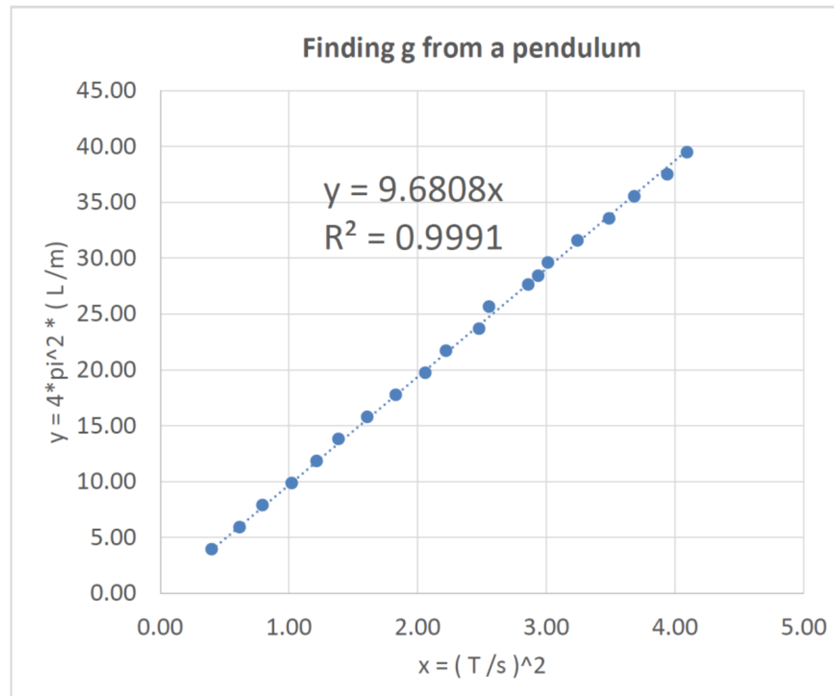
$$\text{SHM:} \quad \ddot{\theta} = -\left(\frac{2\pi}{T}\right)^2 \theta = -\omega^2\theta$$

$$\theta(t) = \theta_0 \cos\left(\frac{2\pi t}{T}\right) = \theta_0 \cos\omega t$$

$$\therefore T = 2\pi\sqrt{\frac{l}{g}} \quad T^2 = 4\pi^2\frac{l}{g} \quad g\underbrace{T^2}_{x} = \underbrace{4\pi^2 l}_{y} \quad \text{i.e.} \quad y = gx$$




**Initial angle = 20 degrees**

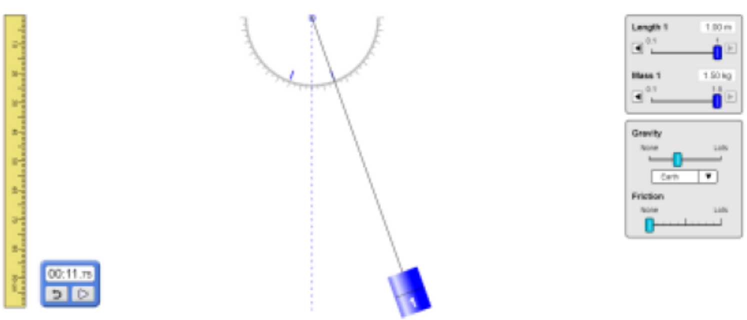


## To complete, underlay (Period vs pendulum length) data with a model curve

$$T = 2\pi\sqrt{\frac{l}{g}}$$

**MEASURING g VIA A PENDULUM** 03/06/2020

| length of pendulum L /m | ten periods /s | ten periods /s | ten periods /s | ten periods /s | period T /s | x = T^2 | y = 4*pi^2 * L |
|---|---|---|---|---|---|---|---|
| 0.1 | 6.25 | 6.25 | 6.45 | 6.32 | 0.63 | 0.40 | 3.95 |
| 0.15 | 7.86 | 7.84 | 7.85 | 7.85 | 0.79 | 0.62 | 5.92 |
| 0.2 | 8.87 | 8.94 | 8.93 | 8.91 | 0.89 | 0.79 | 7.90 |
| 0.25 | 10.07 | 10.14 | 10.11 | 10.11 | 1.01 | 1.02 | 9.87 |
| 0.3 | 11.17 | 10.97 | 10.93 | 11.02 | 1.10 | 1.22 | 11.84 |
| 0.35 | 11.7 | 11.82 | 11.79 | 11.77 | 1.18 | 1.39 | 13.82 |
| 0.4 | 12.67 | 12.69 | 12.69 | 12.68 | 1.27 | 1.61 | 15.79 |
| 0.45 | 13.55 | 13.55 | 13.49 | 13.53 | 1.35 | 1.83 | 17.77 |
| 0.5 | 14.38 | 14.33 | 14.35 | 14.35 | 1.44 | 2.06 | 19.74 |
| 0.55 | 14.94 | 14.86 | 14.91 | 14.90 | 1.49 | 2.22 | 21.71 |
| 0.6 | 15.66 | 15.81 | 15.76 | 15.74 | 1.57 | 2.48 | 23.69 |
| 0.65 | 16 | 15.96 | 16 | 15.99 | 1.60 | 2.56 | 25.66 |
| 0.7 | 16.9 | 16.92 | 16.92 | 16.91 | 1.69 | 2.86 | 27.63 |
| 0.75 | 17.26 | 17.47 | 17.35 | 17.36 | 1.74 | 3.01 | 29.61 |
| 0.8 | 17.95 | 17.97 | 18.11 | 18.01 | 1.80 | 3.24 | 31.58 |
| 0.85 | 18.67 | 18.72 | 18.65 | 18.68 | 1.87 | 3.49 | 33.56 |
| 0.9 | 19.19 | 19.23 | 19.17 | 19.20 | 1.92 | 3.69 | 35.53 |
| 0.95 | 19.89 | 19.85 | 19.82 | 19.85 | 1.99 | 3.94 | 37.50 |
| 1 | 20.24 | 20.23 | 20.24 | 20.24 | 2.02 | 4.10 | 39.48 |
| 0.72 | 17.11 | 17.14 | 17.17 | 17.14 | 1.71 | 2.94 | 28.42 |



Initial angle = 20 degrees

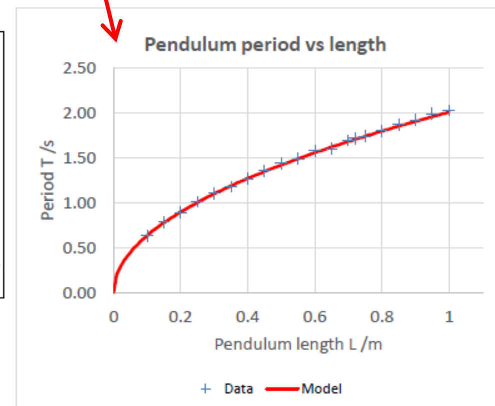
$ml\ddot{\theta} = -mg\sin\theta \approx -mg\theta$ NEWTON II

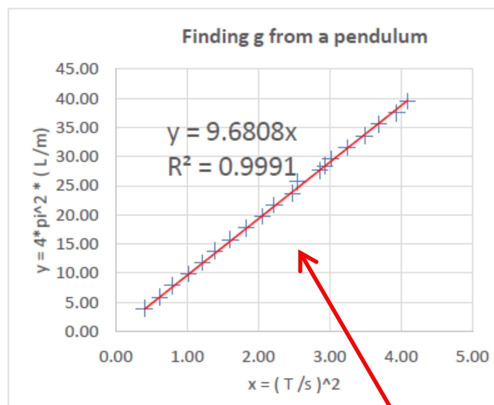
$\therefore \ddot{\theta} = -\frac{g}{l}\theta$

SHM: $\ddot{\theta} = -\left(\frac{2\pi}{T}\right)^2 \theta = -\omega^2\theta$

$\theta(t) = \theta_0 \cos\left(\frac{2\pi t}{T}\right) = \theta_0 \cos\omega t$

$\therefore T = 2\pi\sqrt{\frac{l}{g}} \quad T^2 = 4\pi^2\frac{l}{g} \quad g\underbrace{T^2}_{x} = \underbrace{4\pi^2 l}_{y}$ i.e. $y = gx$




Using g = 9.81N/kg

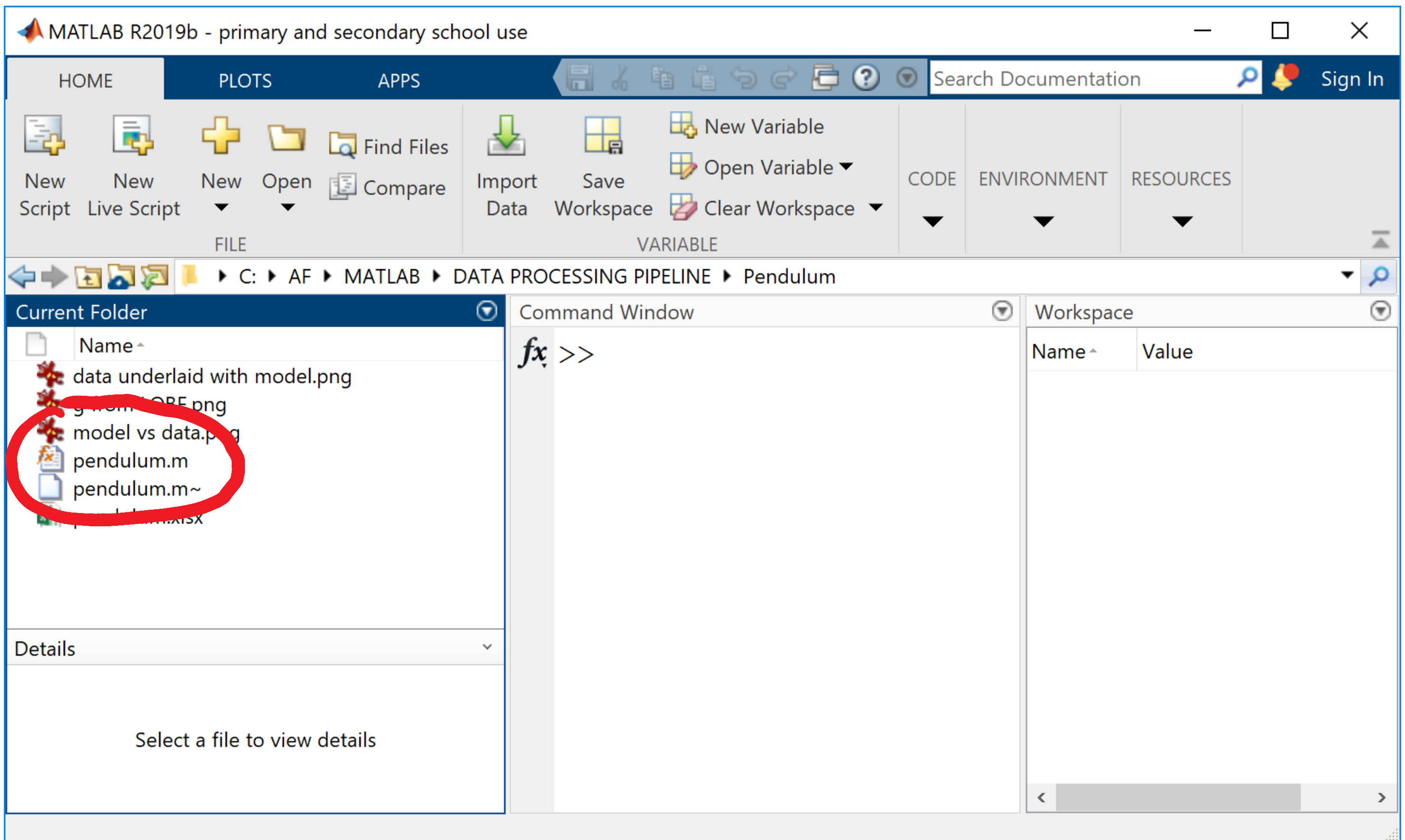
**MODEL curve**

| L /m | T /s |
|---|---|
| 0.000 | 0.000 |
| 0.010 | 0.201 |
| 0.020 | 0.284 |
| 0.030 | 0.347 |
| 0.040 | 0.401 |
| 0.050 | 0.449 |
| 0.060 | 0.491 |
| 0.070 | 0.531 |
| 0.080 | 0.567 |
| 0.090 | 0.602 |
| 0.100 | 0.634 |
| 0.110 | 0.665 |
| 0.120 | 0.695 |
| 0.130 | 0.723 |
| 0.140 | 0.751 |
| 0.150 | 0.777 |
| 0.160 | 0.802 |
| 0.170 | 0.827 |
| 0.180 | 0.851 |
| 0.190 | 0.874 |
| 0.200 | 0.897 |
| 0.210 | 0.919 |
| 0.220 | 0.941 |
| 0.230 | 0.962 |
| 0.240 | 0.983 |
| 0.250 | 1.003 |
| 0.260 | 1.023 |
| 0.270 | 1.042 |
| 0.280 | 1.062 |
| 0.290 | 1.080 |
| 0.300 | 1.099 |
| 0.310 | 1.117 |
| 0.320 | 1.135 |
| 0.330 | 1.152 |
| 0.340 | 1.170 |
| 0.350 | 1.187 |
| 0.360 | 1.204 |
| 0.370 | 1.220 |

Linearization, line of best fit to assess model correlation, and determine *g*

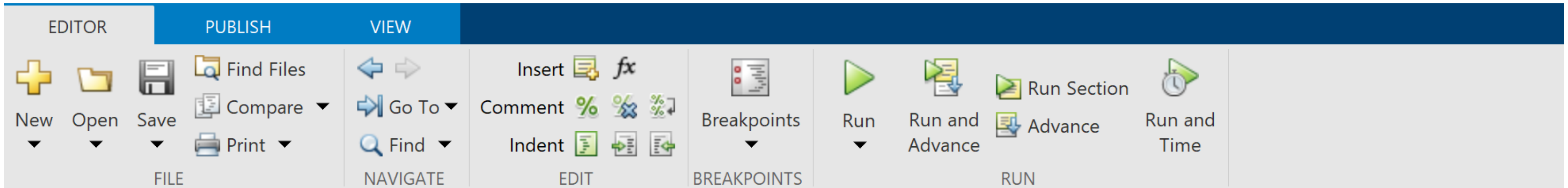
Run **`pendulum.m`** (right click, **run**) to execute a series of commands which constitute the rest of the data processing pipeline. The code can be modified for different experiments.

The key feature is that the code performs the process *automatically*, which can save considerable time when working on new data sets. MATLAB has the ability to perform useful analysis and create bespoke plots to a much higher standard than Excel. Students can focus on the *process,* in modifying the code, rather than the faff of dealing with Excel's defaults! However, I would always start with Excel as a first IT-based analysis.




**MATLAB data processing pipeline**

Editor - C:\AF\MATLAB\DATA PROCESSING PIPELINE\Pendulum\pendulum.m




pendulum.m

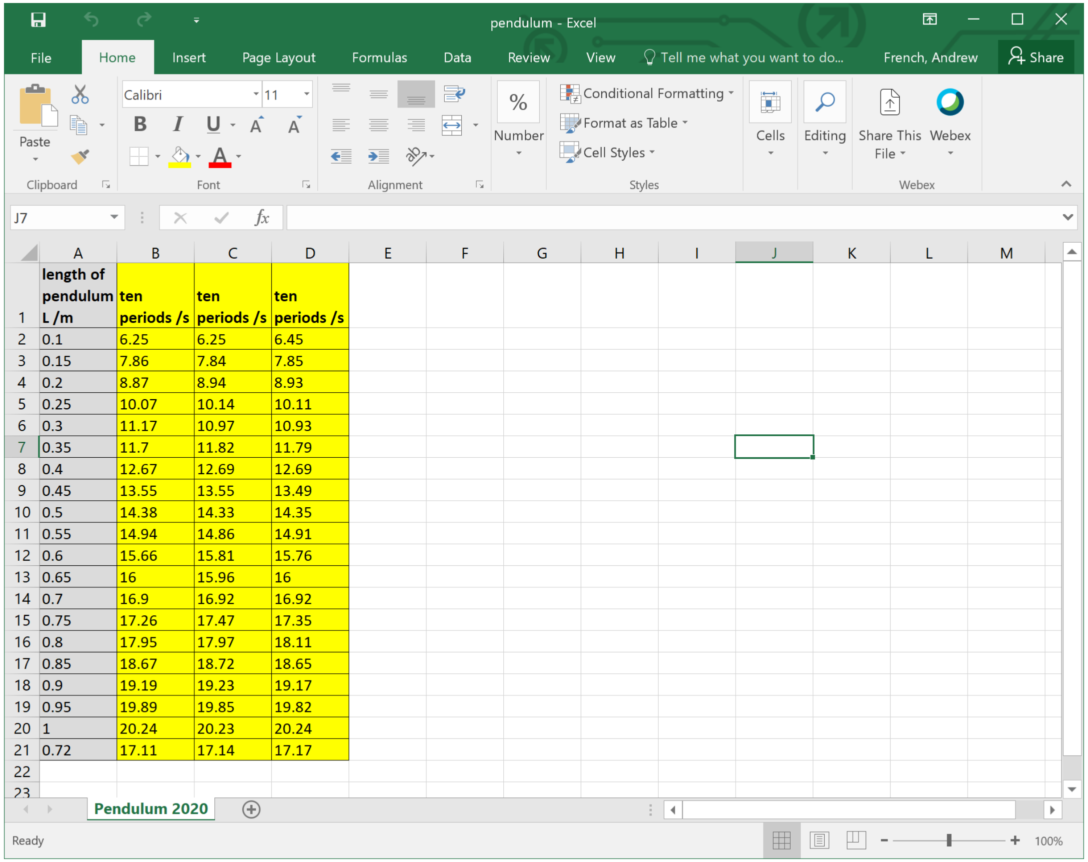
```
% Example physics data processing pipeline: #1 Pendulum
% * Load raw data from an Excel sheet pendulum.xlsx. This has columns of
%   pendulum length L /m, and three repeats of ten periods (10*T) /s.
% * Determine averages and errors
% * Plot y = 4*pi^2 * L vs x =T^2. Determine line of best fit (LOBF) and error,
%   and hence determine g from data. Compare to g = 9.81N/kg.
% * Plot T (data) vs 2*pi*sqrt(L/g) (with actual g). Perform LOBF.
% * Underlay T vs L data and underlay with T = 2*pi*sqrt(L/g) model.
%
% LAST UPDATED by Dr Andrew French. July 2020.

function pendulum

%% INPUTS %%

%Fontsize and marker size for graphs
fsize = 18; msize = 18;

%Set (fixed) error (in m) for pendulum length. Assume no systematic error.
Lerror = 0.01;

%Actual value of g /Nkg^-1
g = 9.81;

%Leave figures or close after printing?
close_after_print = 1;
```

Inside **pendulum.m**

…. It is a text file!

% means commentary

- Vital for humans
- Ignored by machines



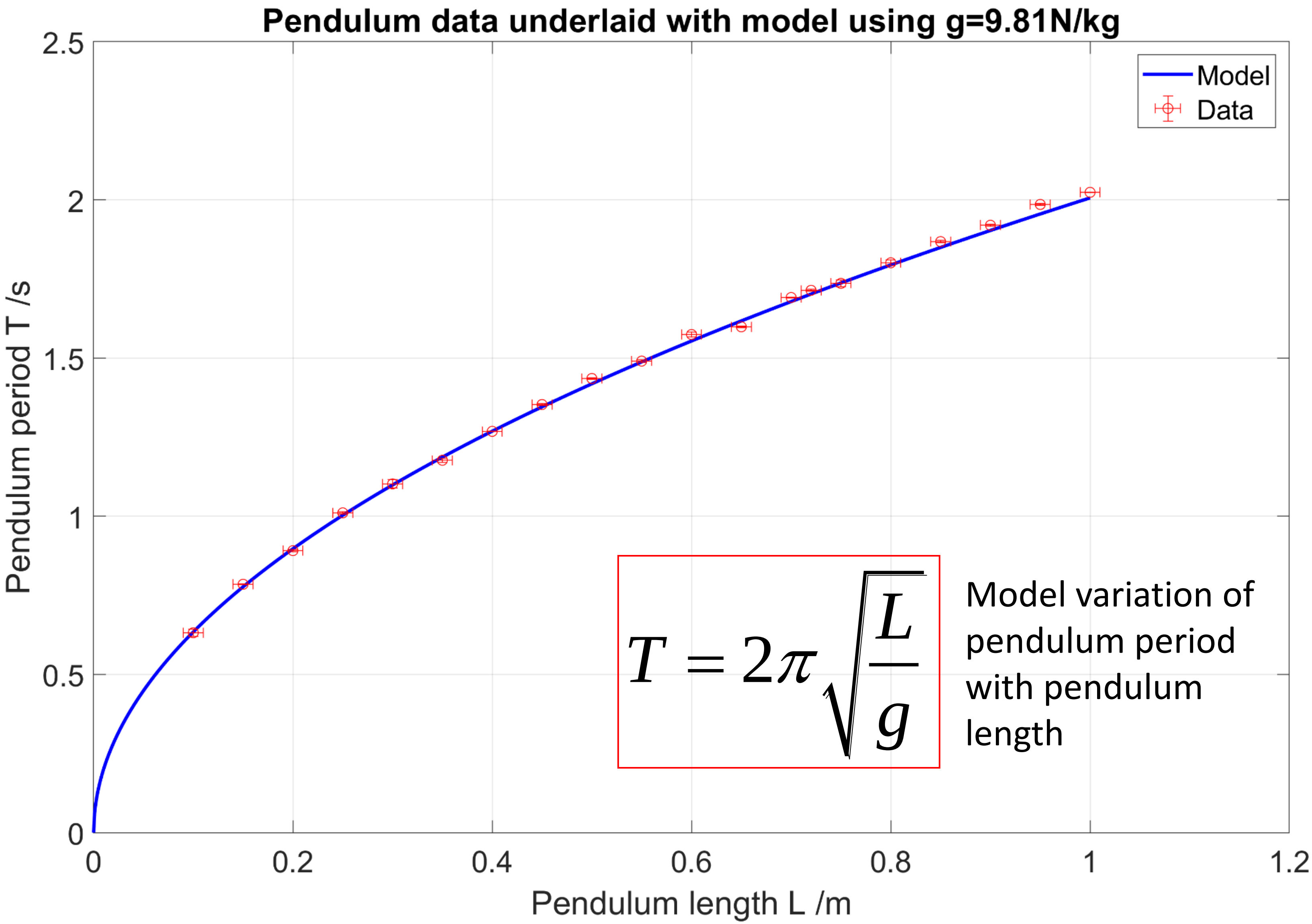
| length of pendulum L /m | ten periods /s | ten periods /s | ten periods /s |
|---|---|---|---|
| 0.1 | 6.25 | 6.25 | 6.45 |
| 0.15 | 7.86 | 7.84 | 7.85 |
| 0.2 | 8.87 | 8.94 | 8.93 |
| 0.25 | 10.07 | 10.14 | 10.11 |
| 0.3 | 11.17 | 10.97 | 10.93 |
| 0.35 | 11.7 | 11.82 | 11.79 |
| 0.4 | 12.67 | 12.69 | 12.69 |
| 0.45 | 13.55 | 13.55 | 13.49 |
| 0.5 | 14.38 | 14.33 | 14.35 |
| 0.55 | 14.94 | 14.86 | 14.91 |
| 0.6 | 15.66 | 15.81 | 15.76 |
| 0.65 | 16 | 15.96 | 16 |
| 0.7 | 16.9 | 16.92 | 16.92 |
| 0.75 | 17.26 | 17.47 | 17.35 |
| 0.8 | 17.95 | 17.97 | 18.11 |
| 0.85 | 18.67 | 18.72 | 18.65 |
| 0.9 | 19.19 | 19.23 | 19.17 |
| 0.95 | 19.89 | 19.85 | 19.82 |
| 1 | 20.24 | 20.23 | 20.24 |
| 0.72 | 17.11 | 17.14 | 17.17 |

**Import** into MATLAB. Assign spreadsheet columns to arrays e.g. `x`, `y`...

```
%% IMPORT EXCEL DATA & PREPARE L, T arrays %%

%Import data. Four columns. First is pendulum length, next three are
% ten periods /s.
[num,txt,raw] = xlsread( 'pendulum' );
L = num(:,1); T10_1 = num(:,2); T10_2 = num(:,3); T10_3 = num(:,4);

%Determine period T /s and the (unbiased estimator) of the error in T.
%The second argument of the std function uses the /(N-1) normalization
T = mean( [T10_1 , T10_2 , T10_3 ],2 )/10;
Terror = std( [T10_1 , T10_2 , T10_3 ],0,2 )/10;
```

```
%% ANALYSIS: Compare T vs L data to model T(L) with actual g %%

%Determine model prediction of T using actual value of g
Tmodel = 2*pi*sqrt( L/g );

%Determine model at a much finer grid of L values
LL = linspace( 0,max(L),1000 ); TTmodel = 2*pi*sqrt( LL/g );

%Plot model curve of T vs L
figure('name','model vs data','color',[1 1 1],...
    'units','normalized','position',[0.05, 0.05, 0.9, 0.85]);
plot( LL, TTmodel,'b-','linewidth',2 ); hold on;
set( gca, 'fontsize',fsize ); grid on;

%Plot data error bars
x = L; y = T; yneg = Terror; ypos = Terror;
xneg = Lerror*ones(size(L)); xpos = Lerror*ones(size(L));
errorbar( x,y,yneg,ypos,xneg,xpos,'o','color','r');

%Graph labels etc
xlabel('Pendulum length L /m'); ylabel('Pendulum period T /s');
title('Pendulum data underlaid with model using g=9.81N/kg');
legend({'Model','Data'});

%Print a PNG file
print( gcf, 'data underlaid with model.png','-r300','-dpng' );
if close_after_print==1; close(gcf); end
```

**Plot** data + error bars, *underlaid* with model curve



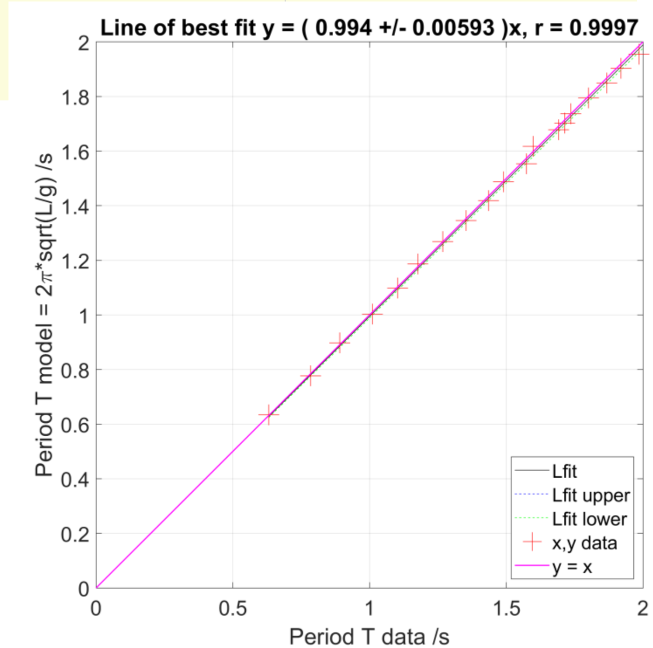

$$T = 2\pi\sqrt{\frac{L}{g}}$$

```
%% ANALYSIS: Determine line of best fit of the form y = m*x between T data and T model
% For 100% correlation, the gradient m = 1 and product-moment correlation coefficient r = 1.
y = Tmodel; x = T; [yfit,xfit,r,m,dm,yupper,ylower,s] = bestfit( x,y );

%Plot line of best fit
xlabel_str = 'Period T data /s';
ylabel_str = 'Period T model = 2\pi*sqrt(L/g) /s';
plot_LOBF( x,y, yfit,xfit,r,m,dm,yupper,ylower,...
    fsize, msize, xlabel_str, ylabel_str );

%Plot y = x for visual check
plot( [0;x], [0;x], 'm-', 'linewidth', 1 );
legend({'Lfit', 'Lfit upper','Lfit lower','x,y data','y = x'},...
    'location','southeast'); axis equal; axis tight;

%Set sensible x,y limits to include origin
xlimits = get(gca,'xlim'); set( gca, 'xlim',[0,round( xlimits(2) )] );
ylimits = get(gca,'ylim'); set( gca, 'ylim',[0,round( ylimits(2) )] );
print( gcf, 'model vs data.png','-r300','-dpng' );
if close_after_print==1; close(gcf); end
```

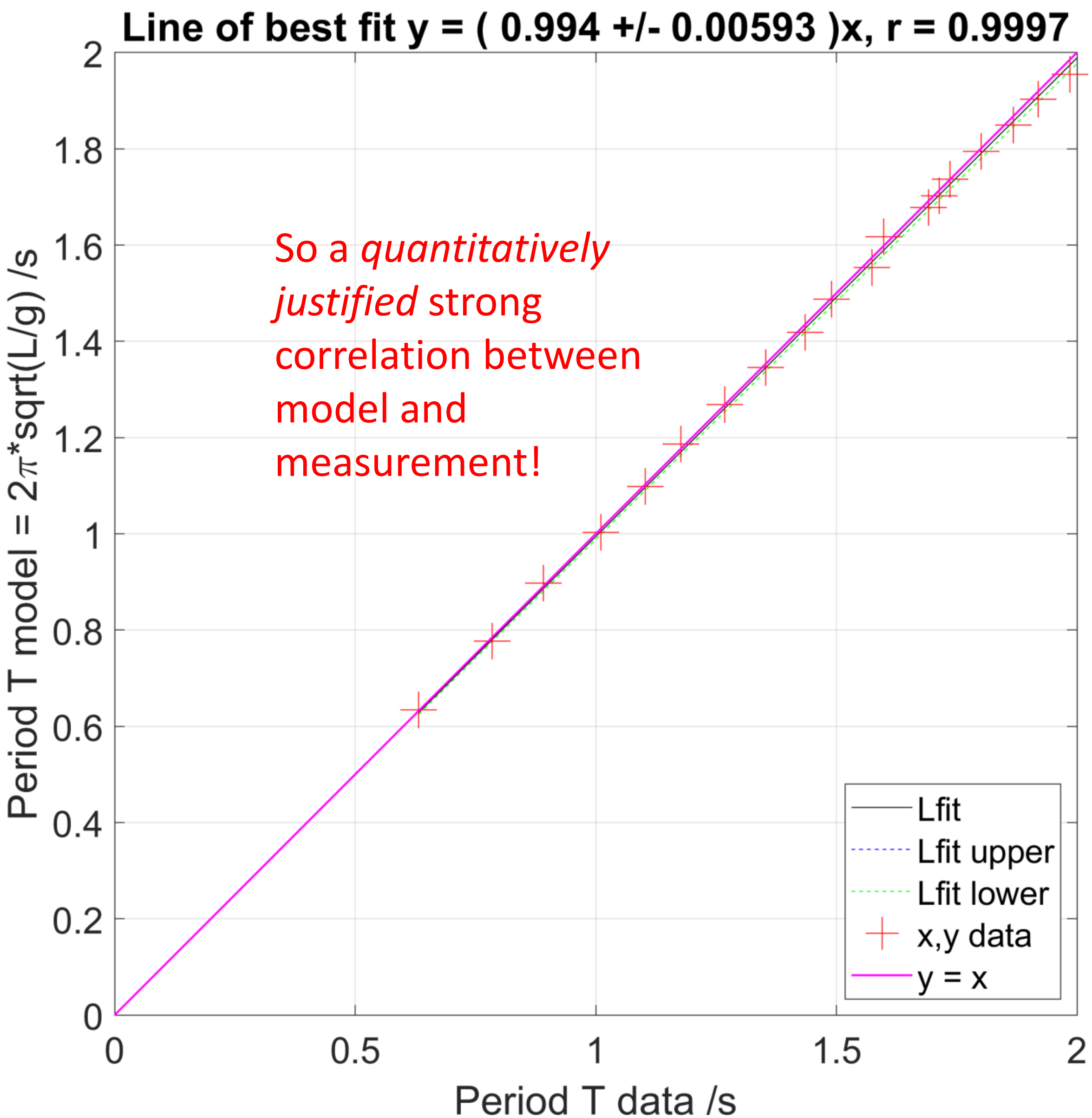
These are *sub-functions* which perform the line of best fit and associated plots. They should be generic, regardless of the dataset.

$$T = 2\pi\sqrt{\frac{L}{g}}$$

Model variation of pendulum period $T$ with pendulum length $L$



**Plot** data vs model
i.e. a $y = x$ graph and
Perform $y = mx$ line of best fit




$$T = 2\pi\sqrt{\frac{L}{g}}$$

Model variation of pendulum period with pendulum length

If you don't need to find parameters from data, **simply comparing model vs measurement** is a very clear first quantitative analysis

```
%% ANALYSIS: Determine g from data %%

%Determine y = 4*pi^2*L and x = T^2
x = T.^2; y = 4*pi^2 * L;

%Determine upper and lower values for error bar calculation
x_upper = ( T + Terror ).^2;  x_lower = ( T - Terror ).^2;
y_upper = 4*pi^2 *( L + Lerror );  y_lower = 4*pi^2 *( L - Lerror );

% Determine line of best fit of the form y = m*x.
% Gradient m is g in this case
[yfit,xfit,r,m,dm,yupper,ylower,s] = bestfit(x,y);

%Plot line of best fit
xlabel_str = '(T/s)^2'; ylabel_str = '4\pi^2*(L/m)';
plot_LOBF( x,y, yfit,xfit,r,m,dm,yupper,ylower,...
    fsize, 0.001, xlabel_str, ylabel_str );

%Plot what the line should be, given the actual value of g
plot( x, g*x, 'm-','linewidth',1 );

%Plot data error bars
yneg = y - y_lower; ypos = y_upper - y; xneg = x - x_lower; xpos = x_upper - x;
errorbar( x,y,yneg,ypos,xneg,xpos,'o','color','r');

%Add a legend
legend({'Lfit', 'Lfit upper','Lfit lower','',...
    'Using g=9.81N/kg','x,y data'},'location','southeast' )
```

But if you *do* need to find parameters, **linearize**, and then perform a line of best fit

$$T = 2\pi\sqrt{\frac{L}{g}}$$

$$\therefore \underbrace{4\pi^2 L}_{y} = g \underbrace{T^2}_{x}$$

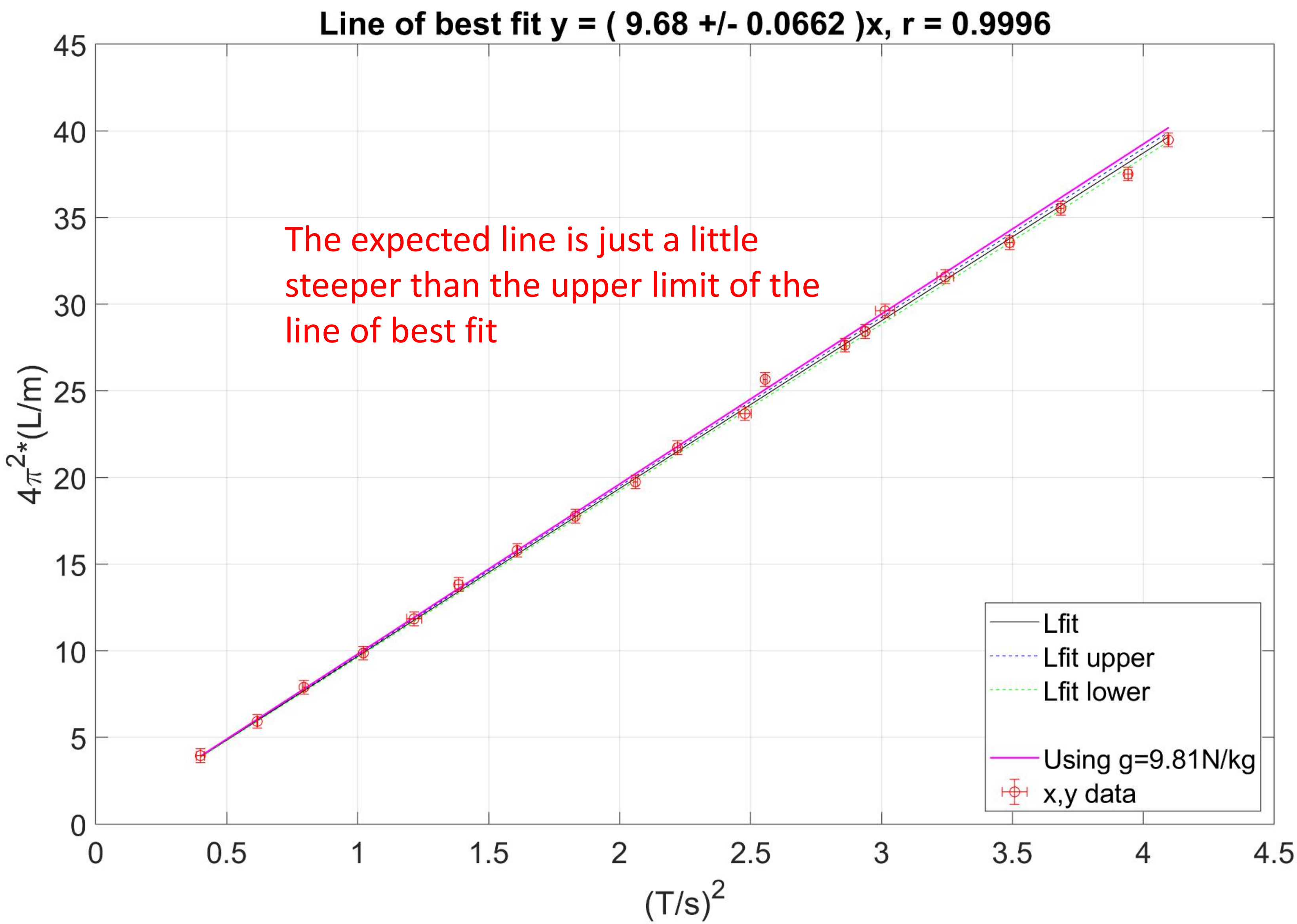
$$\Rightarrow y = gx$$

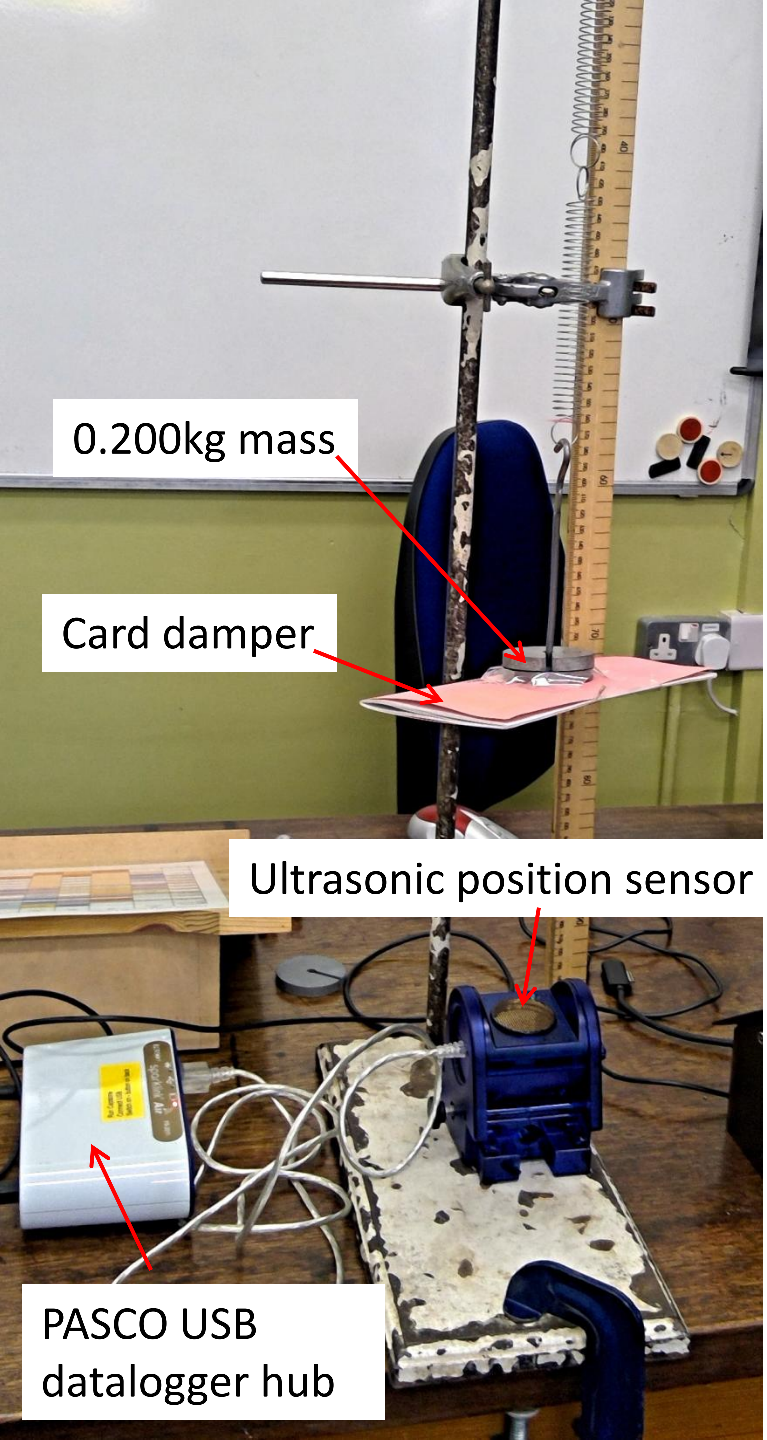
So $g$ is the gradient of the $x,y$ graph in our case

**Plot** linearized graph and use to determine model parameters from gradient (and intercept if $y = mx + c$, not a $y = mx$ fit)

$$T = 2\pi\sqrt{\frac{L}{g}} \quad \therefore \underbrace{4\pi^2 L}_{y} = g \underbrace{T^2}_{x} \quad \Rightarrow y = gx$$

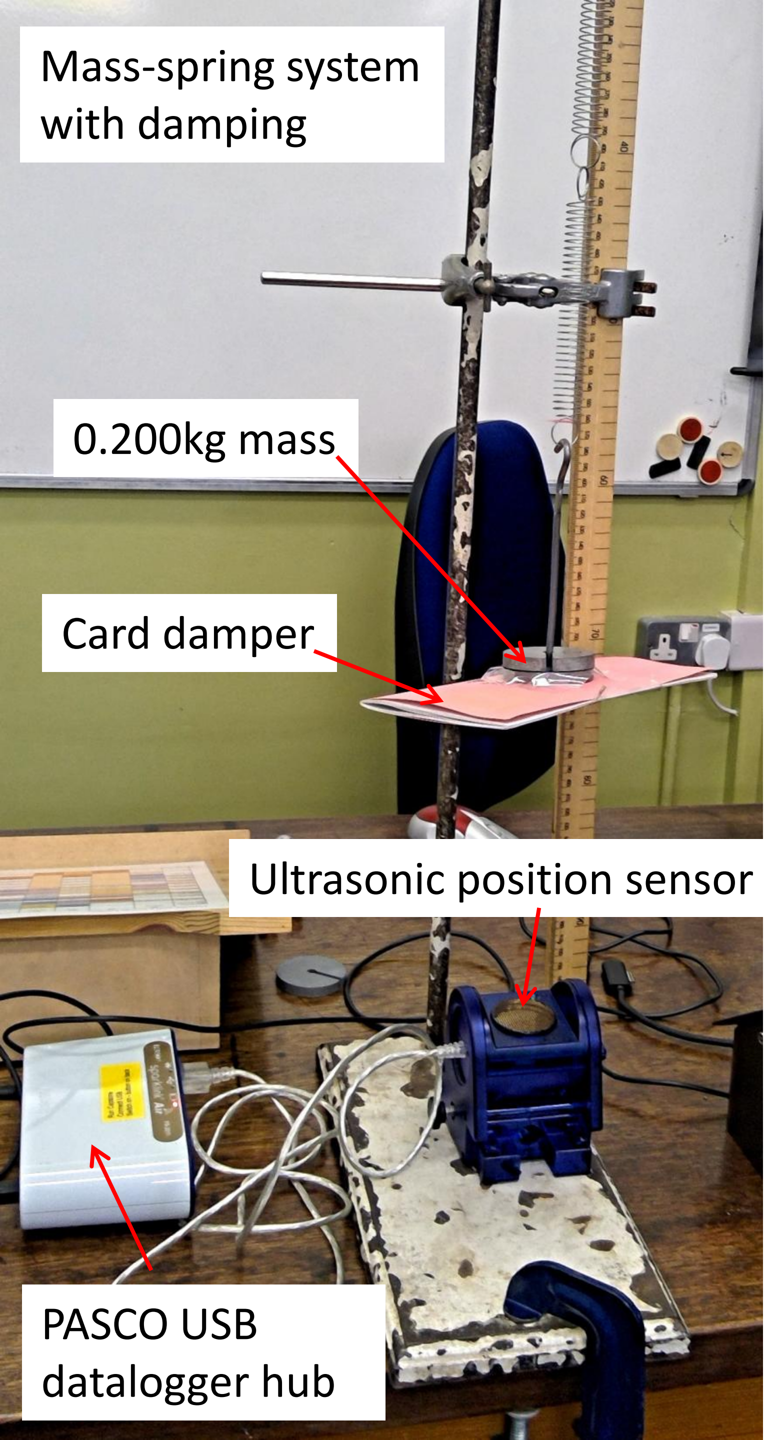
In our case, our gradient (and hence calculated $g$) is systematically lower than what it should be.

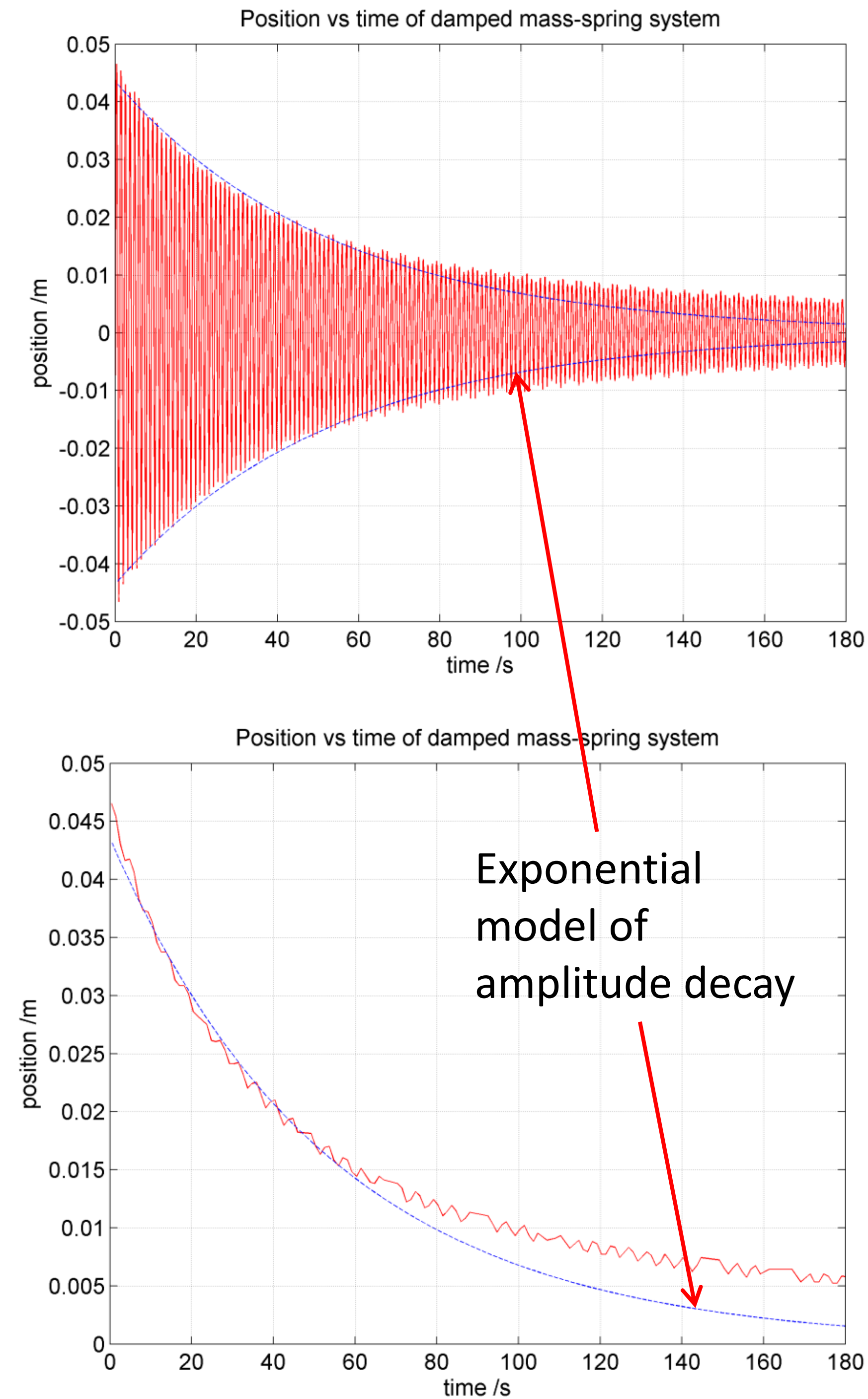


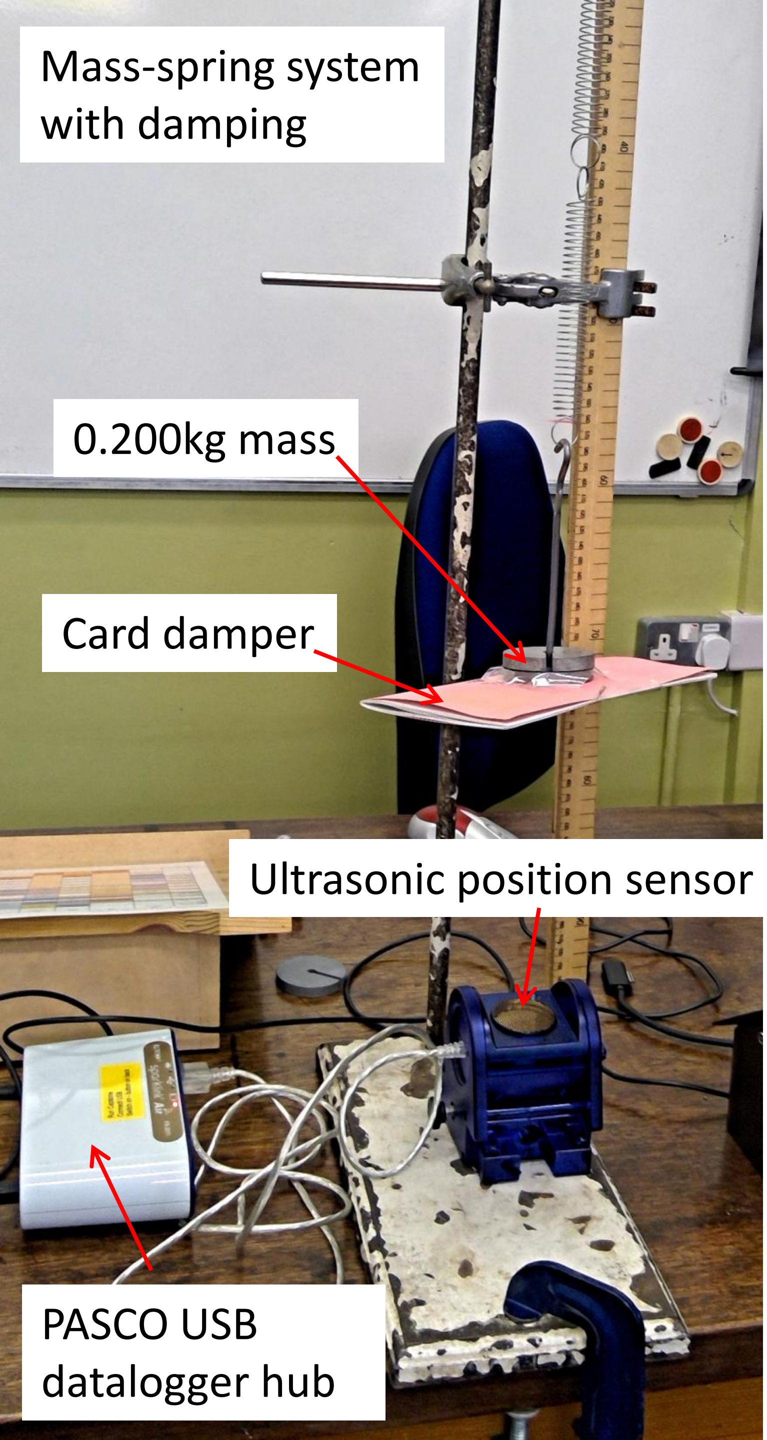


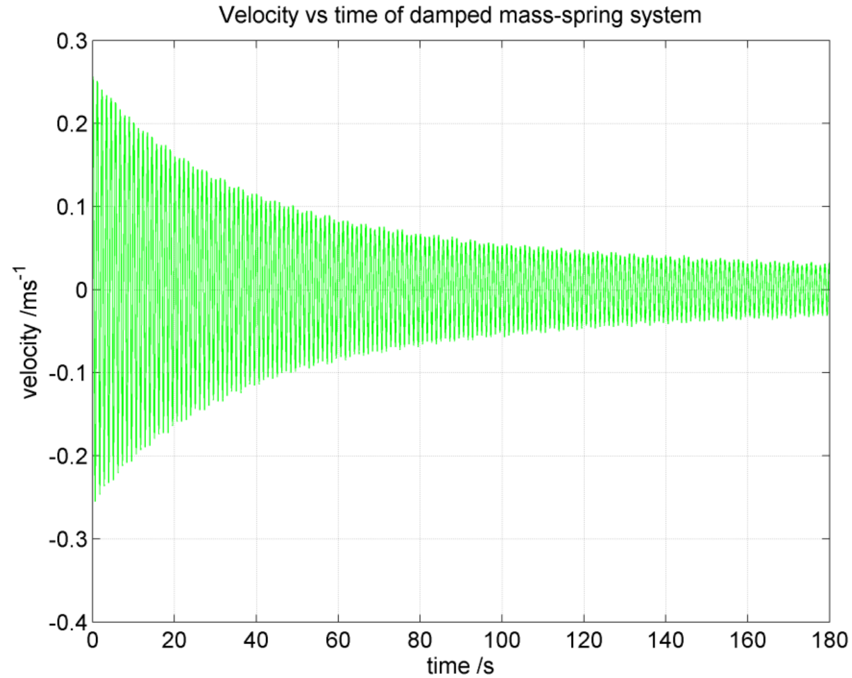

Excel to MATLAB data processing pipeline example:

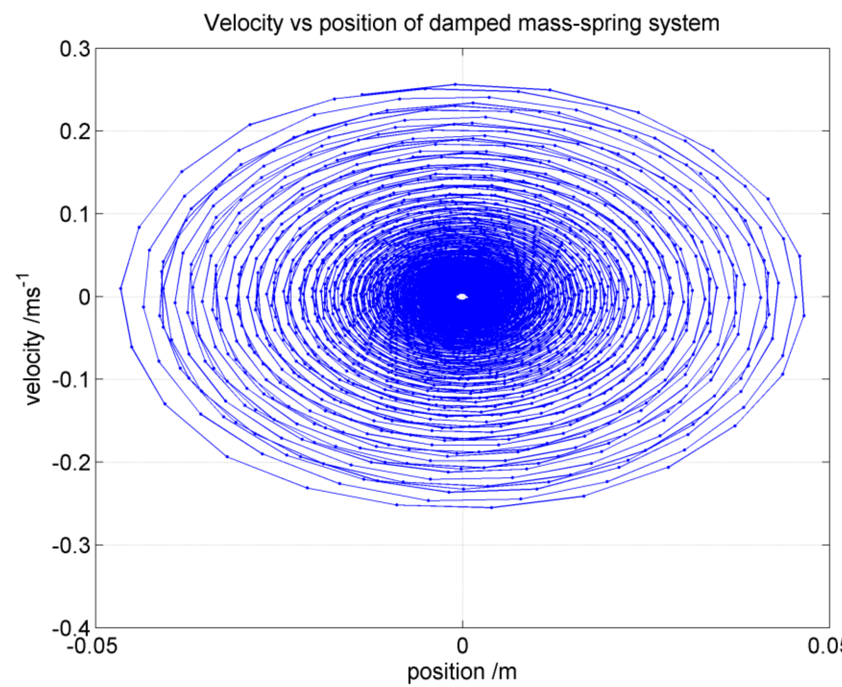
A **mass-spring system** with **damping**, with position recorded via
an ultrasonic sensor and a datalogger.

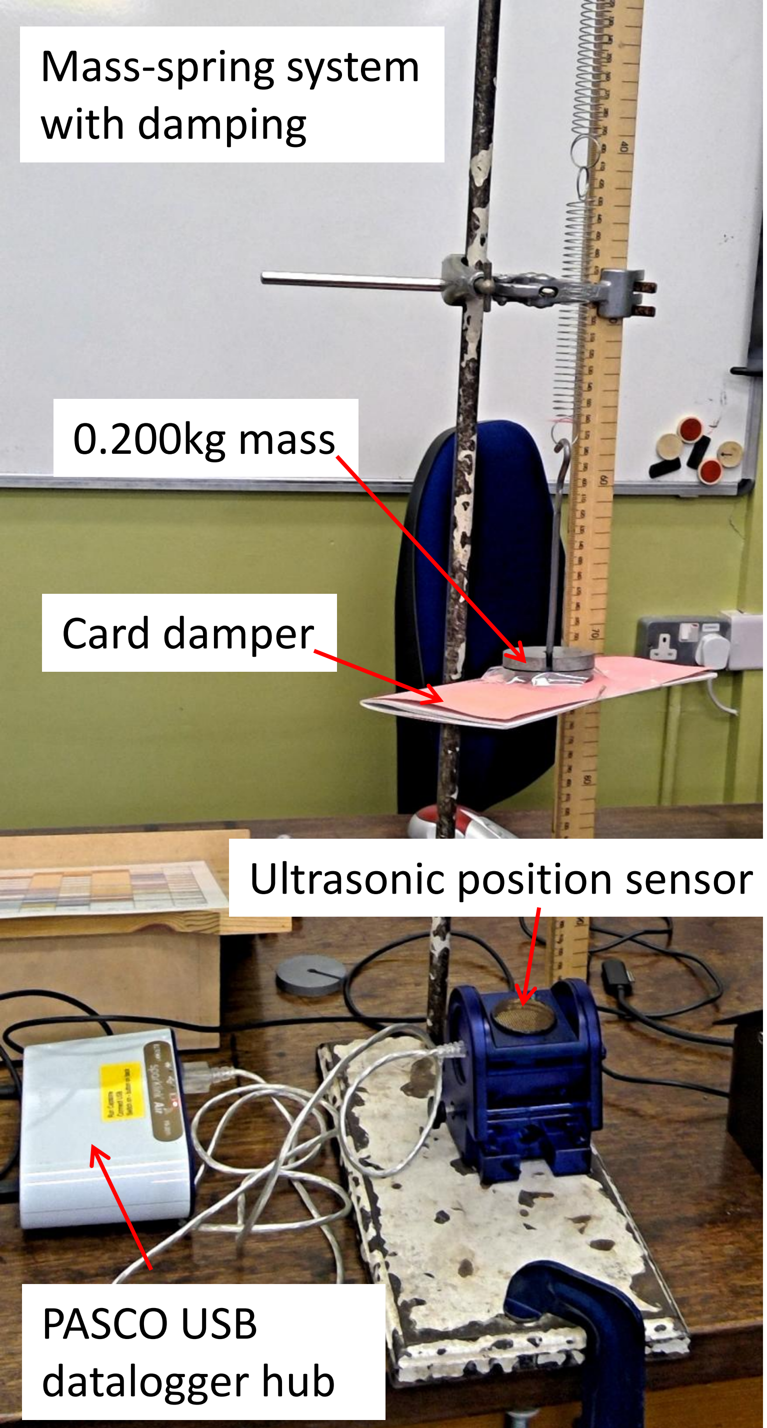

Mass-spring system with damping
0.200kg mass
Card damper
Ultrasonic position sensor
PASCO USB datalogger hub

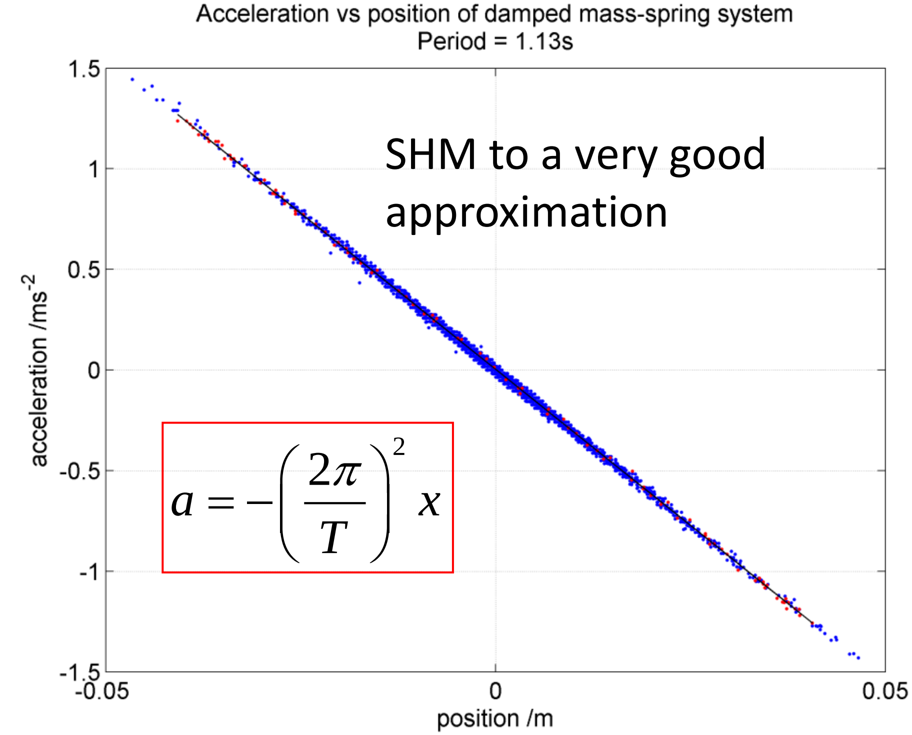

Position vs time of damped mass-spring system
position /m
time /s
0.05
0.04
0.03
0.02
0.01
0
-0.01
-0.02
-0.03
-0.04
-0.05
0
20
40
60
80
100
120
140
160
180
Position vs time of damped mass-spring system
position /m
time /s
0.05
0.045
0.04
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
Exponential model of amplitude decay

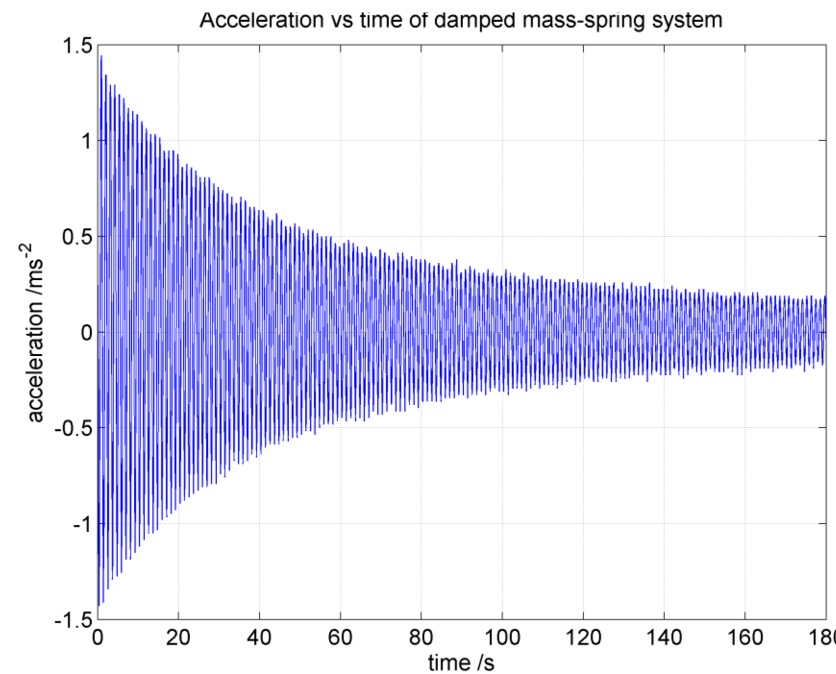

Mass-spring system with damping
0.200kg mass
Card damper
Ultrasonic position sensor
PASCO USB datalogger hub


Velocity vs time of damped mass-spring system
velocity /ms$^{-1}$
time /s
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
0
20
40
60
80
100
120
140
160
180


Velocity vs position of damped mass-spring system
velocity /ms$^{-1}$
position /m
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.05
0
0.05


Mass-spring system with damping
0.200kg mass
Card damper
Ultrasonic position sensor
PASCO USB datalogger hub

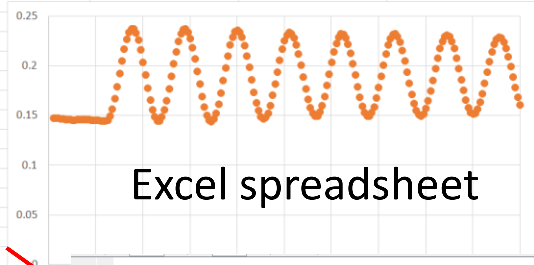

Acceleration vs position of damped mass-spring system
Period = 1.13s
SHM to a very good approximation
$a = -\left(\frac{2\pi}{T}\right)^2 x$
acceleration /ms$^{-2}$
position /m


Acceleration vs time of damped mass-spring system
acceleration /ms$^{-2}$
time /s

# Data processing pipeline for mass-spring system

Datalogger

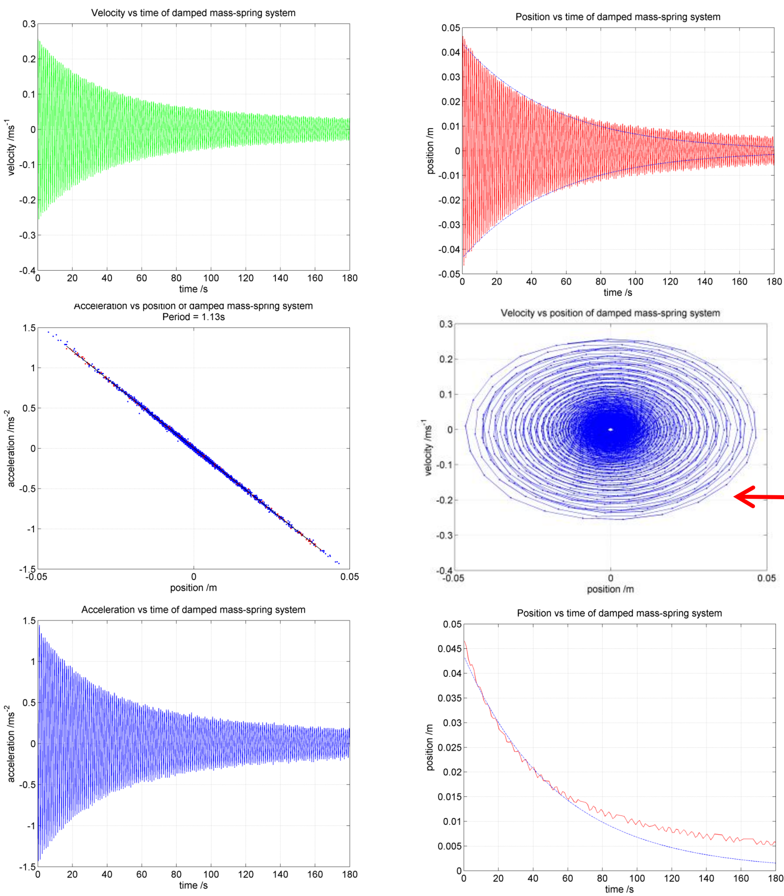
| Position (m) | Velocity (m/s) | Acceleration (m/s²) | Time (s) |
|---|---|---|---|
| 0.1479 | 1.48E-05 | -0.0169 | 0.1 |
| 0.1479 | -0.0017 | -0.0516 | 0.15 |
| 0.1478 | -0.0051 | -0.0519 | 0.2 |
| 0.1474 | -0.0069 | -0.0176 | 0.25 |
| 0.1471 | -0.0069 | 0.0343 | 0.3 |
| 0.1467 | -0.0035 | 0.0522 | 0.35 |
| 0.1467 | -0.0017 | -0.0172 | 0.4 |
| 0.1465 | -0.0052 | -0.0181 | 0.45 |
| 0.1462 | -0.0035 | 0.0861 | 0.5 |
| 0.1462 | 0.0034 | 0.0695 | 0.55 |
| 0.1465 | 0.0035 | -0.0172 | 0.6 |
| 0.1465 | 0.0017 | -0.0173 | 0.65 |
| 0.1467 | 0.0017 | 4.41E-04 | 0.7 |
| 0.1467 | 0.0018 | -0.0512 | 0.75 |
| 0.1469 | -0.0034 | -0.1038 | 0.8 |
| 0.1464 | -0.0086 | -0.0351 | 0.85 |
| 0.146 | -0.0069 | 0.0515 | 0.9 |
| 0.1457 | -0.0035 | 0.0692 | 0.95 |




Excel spreadsheet

MATLAB

```matlab
% Vertically mounted three-springs-in-series are attched to a 0.2kg mass
% and sellotaped to a card diary to increase air resistance.
% Position of the mass is determined via a PASCO ultrasonic transducer,
% and data is logged at 20Hz using the Capstone system. Capstone
% automatically determines velocity and acceleration via some form of
% numerical differentiation.
function damped_mass_on_spring_analysis
fsize = 14;  %Fontsize for graphs
tstart = 0;  tend = 60*3;  %Time interval

%Load data
[num,txt,raw] = xlsread( 'Damped mass on spring.xlsx');

%Ignore first few data points (oscillation hasn't started yet)
num = num(50:end,:);

%Determine position, velocity, acceleration, time in SI units
t = num(:,4); t = t - t(1);
x = num(:,1); v = num(:,2); a = num(:,3);

%Determine equilibrium height from the average position, and shift x
%such that equilibrium displacement is zero.
x = x - mean(x);

%Restricted data set (i.e. where effect of damping can be ignored)
aa = a(100:200); tt = t(100:200); xx = x(100:200);

%Determine line of best fit between a and x to determine period
[m,c,yfit,r] = lfit(xx,aa); T = 2*pi/sqrt(-m);

%Find peaks and plot these to determine exponential envelope
[tp,xp] = peakfindergeneral(t,x); N = length(xp);

%Linearize exponential envelope. Choose which set of data you want to use
%for this computation. NOTE THE ENVELOPE IS NOT EXPONENTIAL IN THE SAME WAY
%DURING THE WHOLE DECAY, SINCE THE DRAG FORCE WON'T BE LINEAR WITH SPEED.
i = find(xp>0); i = i(1:50);
Y = log( xp(i) ); X = tp(i);

%Determine exponential envelope in the form x = (+/-) A*exp(-k*t)
[m,c,YFIT,r] = lfit( X,Y);
```




Auto-generated graphs and analysis

BPhO Computational Challenge

- Suggested homework
- Q&A

BPhO British Physics Olympiad