Head of Class Write-up - Spreadsheet Physics

Matthew Johnson
Trinity Hall

April 10, 2002

Abstract

Head of Class Write-up of the *Spreadsheet Physics* practical done on 7th March 2002. This was a practical exercise, involving the spreadsheet package Microsoft Excel 2000, and worked through various uses for spreadsheets in physics investigations. The practical covered simple calculations, visual display of data, and the use of a spreadsheet in simulations and modelling of physical situations. Also introduced were macros and basic animations that can be produced.

Contents

1 Initial setup

To start of the experiment I logged into a PWF [1] computer, started the Microsoft Excel 2000 spreadsheet program, and created a new document. This was saved on my PWF file-space. Each of the separate parts of the experiment was placed in a different sheet, and I first did all of the compulsory problems, followed by an appropriate selection of the optional exercises. For convenience, however, they are included here in numerical order.

Please note that throughout this experiment I have made use of the *Create Name* facility of Excel to assign identifying names to certain cells, which enables me to refer to them by those names, rather than by cell location. This reference is an absolute address, rather than a relative one (i.e. it is not affected by the movement, or copying of cell contents to another location).

2 Simple Spreadsheet Calculations

2.1 Count, Sum & Mean

I entered the data given in the worksheet for this section, that is a series of data in column A, and the functions `COUNT`, `SUM` and the mean function in columns B & C. You can see this in figure 1.

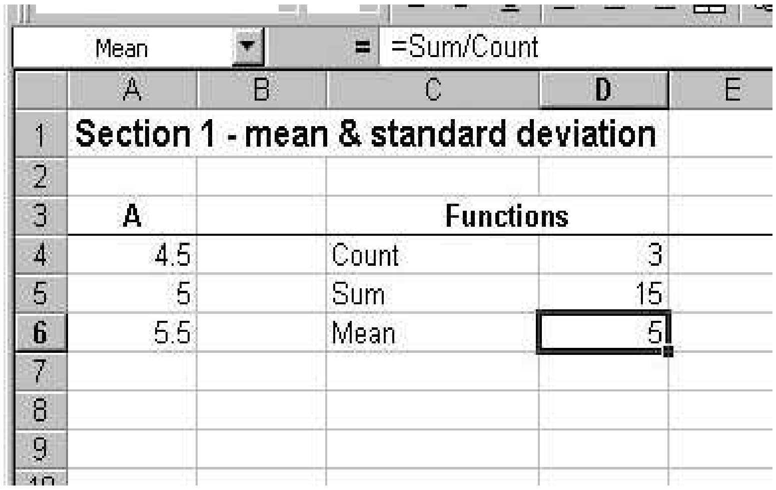


Figure 1: Initial Data

The data set in column A was set the name *A*, and the formulae in column D were then entered as `=COUNT(A)` and `=SUM(A)`. These two values were assigned

[1] Public Workstation Facility

the names *Count* and *Sum*, and then the mean is calculated by `=Sum/Count`, and named *Mean*.

I then increased the size of the dataset A and watched the values of *Count* and *Sum* change. As I had named the dataset A it was not necessary to change the formulae used in the *Count*, *Sum* and *Mean* fields, they updated automatically. Table 1 contains the values of the count and sum as I added the values in the 1st column into the dataset A (whilst keeping the old values).

Extra Values	Count	Sum
4.5, 5, 5.5	3	15
2.4	4	17.4
1.8	5	19.2
5.6	6	24.8
4.8	7	29.6
4.7	8	34.3
5.4	9	39.7
1.3	10	41.0

Table 1: Values entered to test the COUNT & SUM functions

As you can see from that, `COUNT` gives you the number of values in the dataset, and `SUM` totals them up. Hence you can calculate the mean using

$$\bar{x} = \frac{\sum x_i}{n}$$

2.2 Standard Deviation

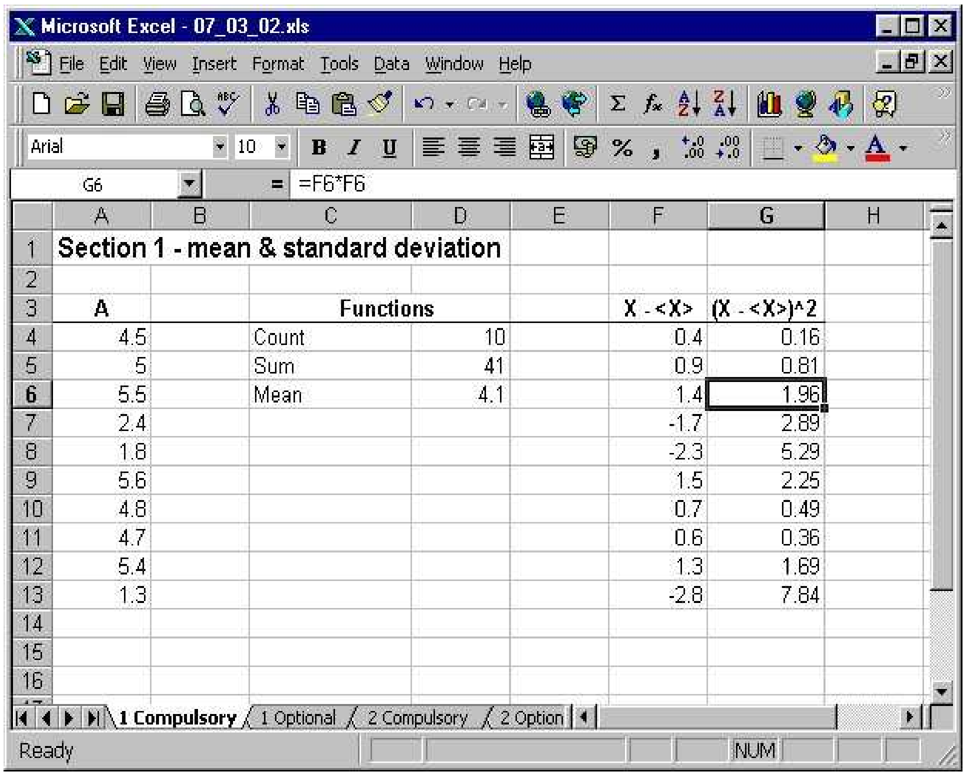
The F column was setup by putting the formula `=A4-Mean` into F4, and filling it down to F13. You can see here how using the name *Mean* rather than the absolute address `$D$6` is much clearer. When the F4 cell was copied down, the values changed from `A4` to the row number it was pasted into (because that is a **relative** address), but `Mean` didn't change, since that is an **absolute** address. This dataset (F4-F13, labelled *diffX*) contains the value $(x - \bar{x})$ for each of the values in A

2.2.1 Addressing

Relative addressing stores the target cell location as an offset relative to the current cell. Hence, when it is moved or copied, the cells it refers to are changed by the same offset it was moved or copied by. Absolute addressing, on the other hand, stores the actual location of the cell, and this stays the same when the formula is moved or copied. Relative addressing is the default in Excel, and absolute addressing is specified by prefixing a `$` to the static part of the location (e.g. `$D$6`). This is also the behaviour when you specify a name to a cell or group of cells.

As a demonstration of this, I copied the formula `=COUNT(A$1:A$10)` from the cell E2 to F1, thus moving it up and right 1 cell. The formula in F1 became `=COUNT(B$1:B$10)`. As you can see, the relative part (the column) changed, while the absolute part (the rows) stayed the same. Doing the same with the formula `=B2 + C$3 + $D4 + $E$5` changes to `=C1 + D$3 + $D3 + $E$5`, as expected - this demonstrates the different combinations of absolute and relative addressing you can do. Also note that if you move (cut & paste) a cell that is being referred to (for example C1 in the above formula), Excel will automatically update the cell that is referring to it.

The G column in the spreadsheet was setup by squaring each value in the F column, thus giving the values of $(x - \bar{x})^2$ for each value in A. This column was labelled as *diff2X*. You can see this in action in figure 2.



	A	B	C	D	E	F	G	H
1	Section 1 - mean & standard deviation							
2								
3	A		Functions			X - <X>	(X - <X>)^2	
4	4.5		Count	10		0.4	0.16	
5	5		Sum	41		0.9	0.81	
6	5.5		Mean	4.1		1.4	1.96	
7	2.4					-1.7	2.89	
8	1.8					-2.3	5.29	
9	5.6					1.5	2.25	
10	4.8					0.7	0.49	
11	4.7					0.6	0.36	
12	5.4					1.3	1.69	
13	1.3					-2.8	7.84	
14								
15								
16								

Figure 2: $(x - \bar{x})$ & $(x - \bar{x})^2$

Lastly, I setup the standard deviation in cell C7, and named it *Sigma*. The formula for this is `=SQRT( SUM(diff2X) / (COUNT(diff2X)-1) )`. The error on the mean (cell D8) can be calculated with `=Sigma/SQRT(Count)` and combining this with the mean gives you the actual result, as shown in figure 3. This is the final state of the worksheet, and in fact the result given at the bottom of the Functions columns uses the built in `AVERAGE`, `COUNT` & `STDEV` functions, operating on the dataset A.

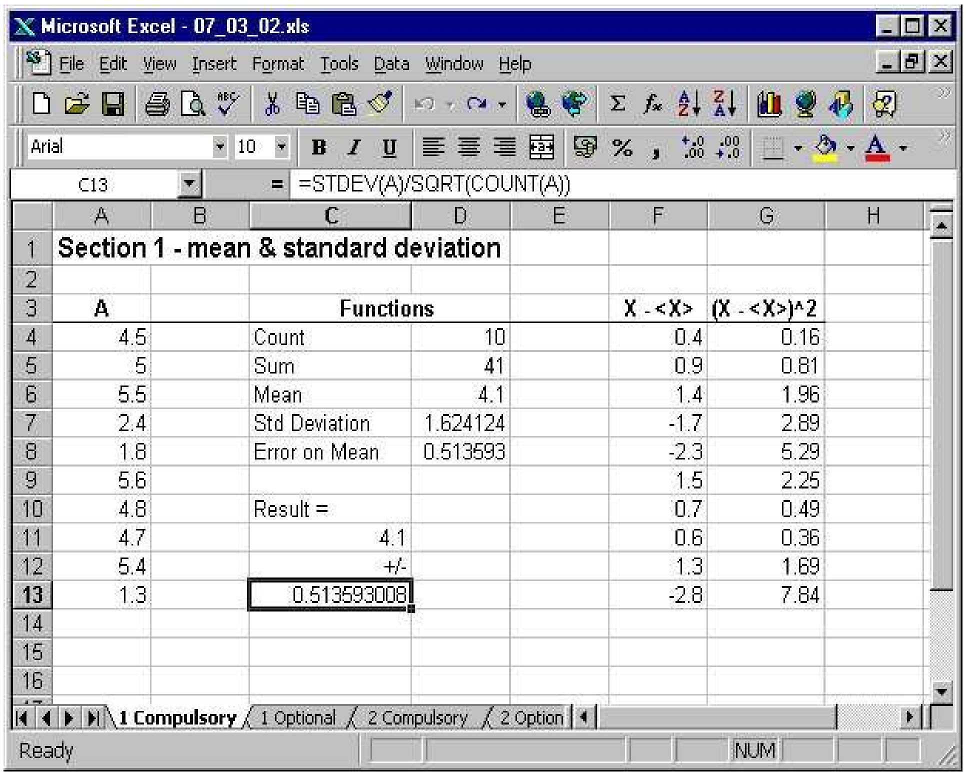


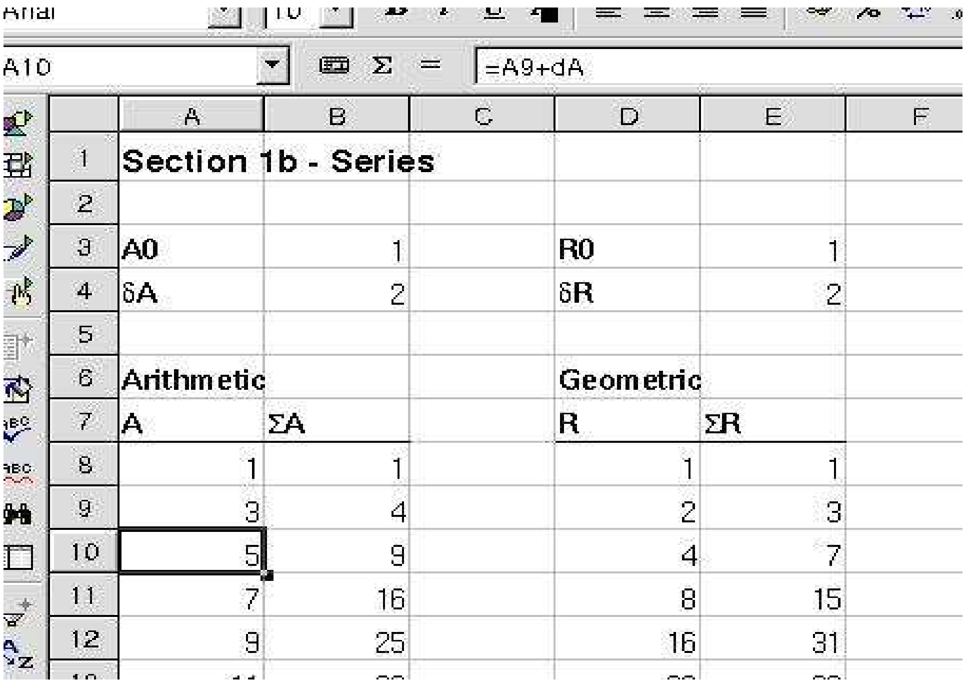
Figure 3: Finally

2.3 Series

This was the first optional exercise, and used the spreadsheet to show the terms and sums of an arithmetic and a geometric progression. The spreadsheet can be seen in figure 4 with the first few values of each series. The formula in the AP terms (from A9) was `=A8 + dA`, and for the GP, `=D8 * dR`. The formulae used for calculating the sum to the nth term used both absolute and relative addressing to specify which area to sum over. For the AP, in cell B11 it was `=SUM($A$8:A11)` - as you can see from this, as that formula is filled down, the first term stays constant, but the second one increases, so that calculates the sum up to that row. Of course you could use the formulae for sums & terms,

$$T_n = a + (n-1)d \quad S_n = \frac{n}{2}(2a + (n-1)d)$$
$$T_n = ar^{(n-1)} \qquad S_n = a\frac{(r^n - 1)}{(r-1)}$$

However, the iterative method suits the spreadsheet, and can be used in a proof of the above formulae.


A10 =A9+dA

	A	B	C	D	E	F
1	Section 1b - Series					
2						
3	A0	1		R0	1	
4	δA	2		δR	2	
5						
6	Arithmetic			Geometric		
7	A	ΣA		R	ΣR	
8	1	1		1	1	
9	3	4		2	3	
10	5	9		4	7	
11	7	16		8	15	
12	9	25		16	31	

Figure 4: Arithmetic & Geometric Series

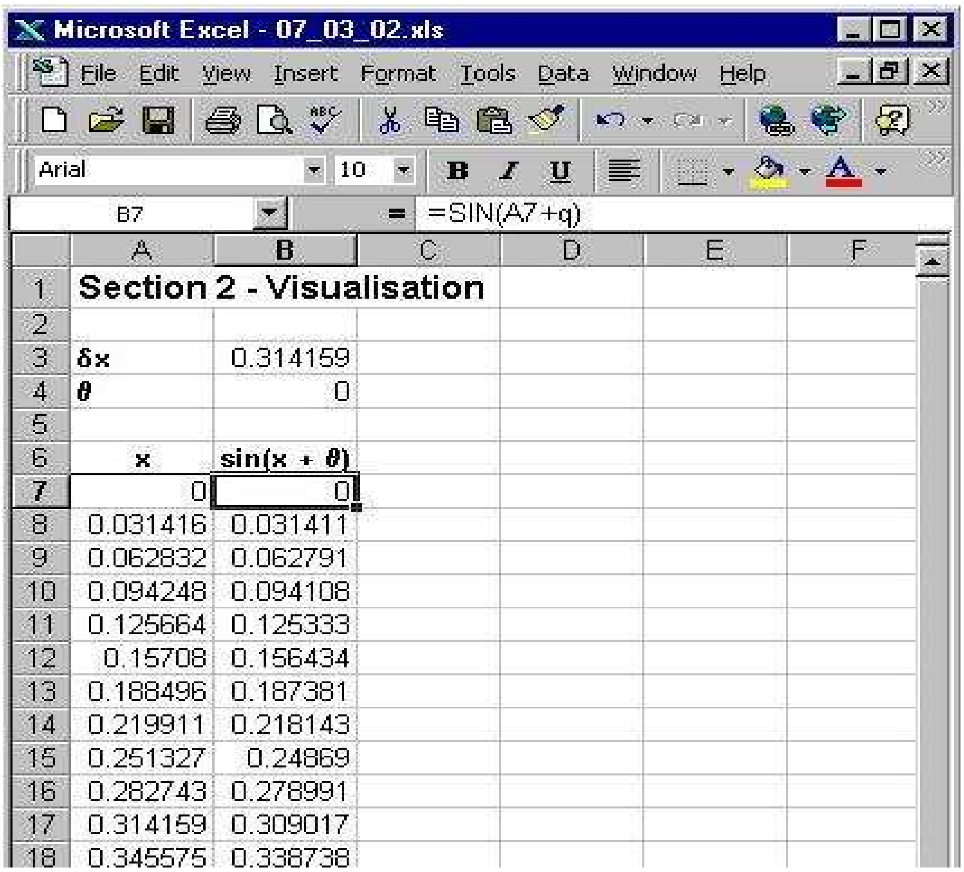
3 Visualisation

3.1 Setup & Formulae

This section demonstrated producing visualisations of functions using the Graph function in Excel. I started by plotting the graph of $\sin(x + \theta)$, where x incremented by δx. To do this I setup the phase and x-increment in cells at the top of the sheet and named them dx and q respectively. I then setup 2 columns, for x and $\sin(x + \theta)$, with the following formulae:

x (column A, starting from A7)	starting at 0 and the other cells filled down as `=A7+dx`
$\sin(x + \theta)$ (column B, starting from B7)	all cells filled down as `=SIN(A8+p)`

δx was initially set to $\pi/10$ and θ to 0. The formulae were filled down to row 34, and you can see some of the values in figure 5.



	A	B	C	D	E	F
1	Section 2 - Visualisation					
2						
3	δx	0.314159				
4	θ	0				
5						
6	x	sin(x + θ)				
7	0	0				
8	0.031416	0.031411				
9	0.062832	0.062791				
10	0.094248	0.094108				
11	0.125664	0.125333				
12	0.15708	0.156434				
13	0.188496	0.187381				
14	0.219911	0.218143				
15	0.251327	0.24869				
16	0.282743	0.278991				
17	0.314159	0.309017				
18	0.345575	0.338738				

Figure 5: Values for $\sin(x + \theta)$

3.2 Charting

Using the Excel Insert Chart facility, I setup a graph of $\sin(x+\theta)$, by specifying the chart data source as x for the x values and sin for the y values (x and sin are names I assigned to the two columns of the worksheet). This produced a graph of $\sin(x+\theta)$, and by changing the value of δx you can change the resolution of the curve displayed, and by changing θ you shift the point at which is crosses the axis (the phase of the wave). Two examples of this with different settings are shown in figure 6.

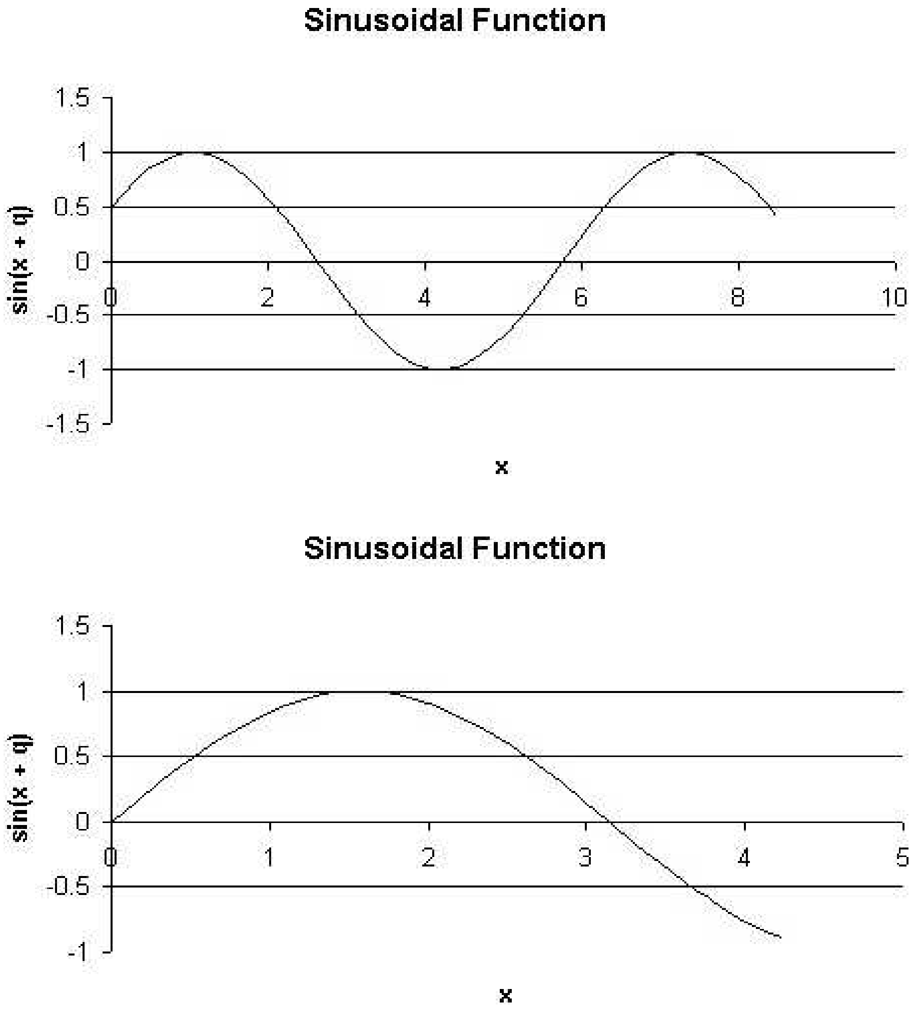


Figure 6: The first graph is with $\delta x = \frac{\pi}{10}$ and $\theta = \frac{\pi}{6}$ and the second has $\delta x = \frac{\pi}{20}$ and $\theta = 0$

3.3 Fourier Series

The optional task I chose to do in this section was the Fourier Series. I started off by copying the sheet from the first visualisation task, but added a second term to the sin column. This gives us a 2 term Fourier series approximating a square wave.

$$f(x) = \sin(x + \theta) + \frac{1}{3}\sin(3(x + \theta)) \tag{1}$$

The phase constant θ is set to 0 and equation 1 is entered into the second column as the Excel function `=SIN(A7 + q) + 0.333*SIN(3*(A7 + q))` where q is the name for the phase offset θ cell. This is again filled down into the whole dataset named *Fourier*. I then created a chart from that data, and as you can see in figure 7 it is only an approximation to a square wave.

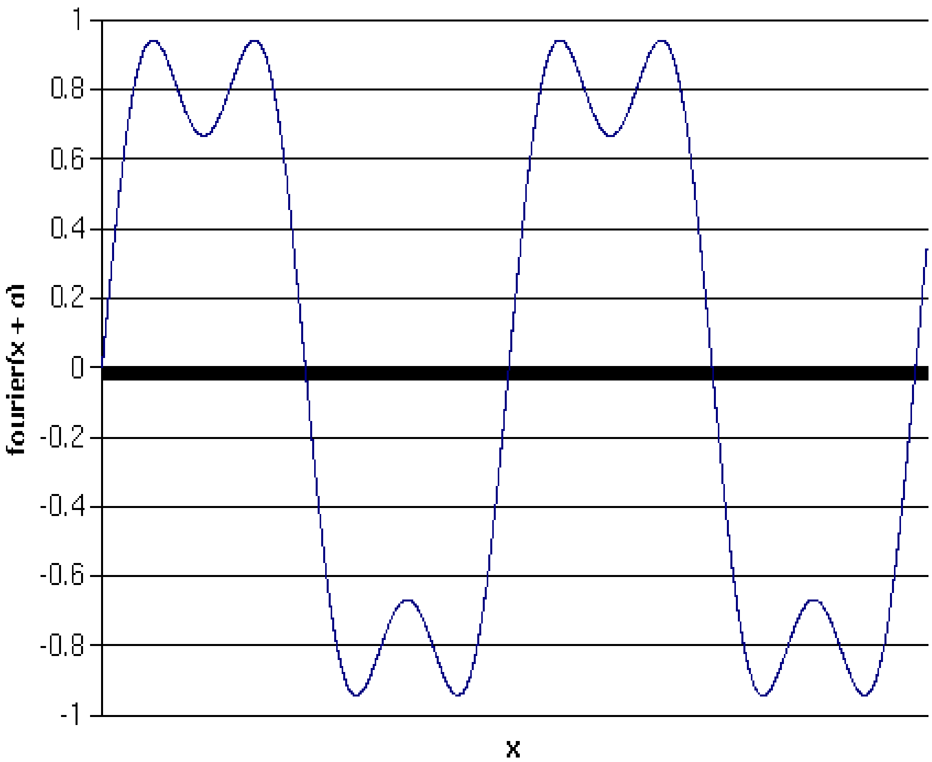


Figure 7: 2 term Fourier Series approximation

I first tried to improve the approximation by increasing the resolution (by decreasing δx), and increasing the number of terms calculated so that it covers the same range, but that didn't have the desired effect. To get a better approximation to a square wave you would have to add another term to the Fourier series used in the calculation. I tried changing equation 1 to

$$f(x) = \sin(x + \theta) + \frac{1}{3}\sin(3(x + \theta)) + \frac{1}{5}\sin(5(x + \theta)) \tag{2}$$

This gives a better approximation, and the 4 term Fourier series shown in figure 8 is better still. Note that the coefficients of even n are 0, so the 4th term actually has $n = 7$ for the coefficients of $\frac{1}{n}\sin(nx)$.

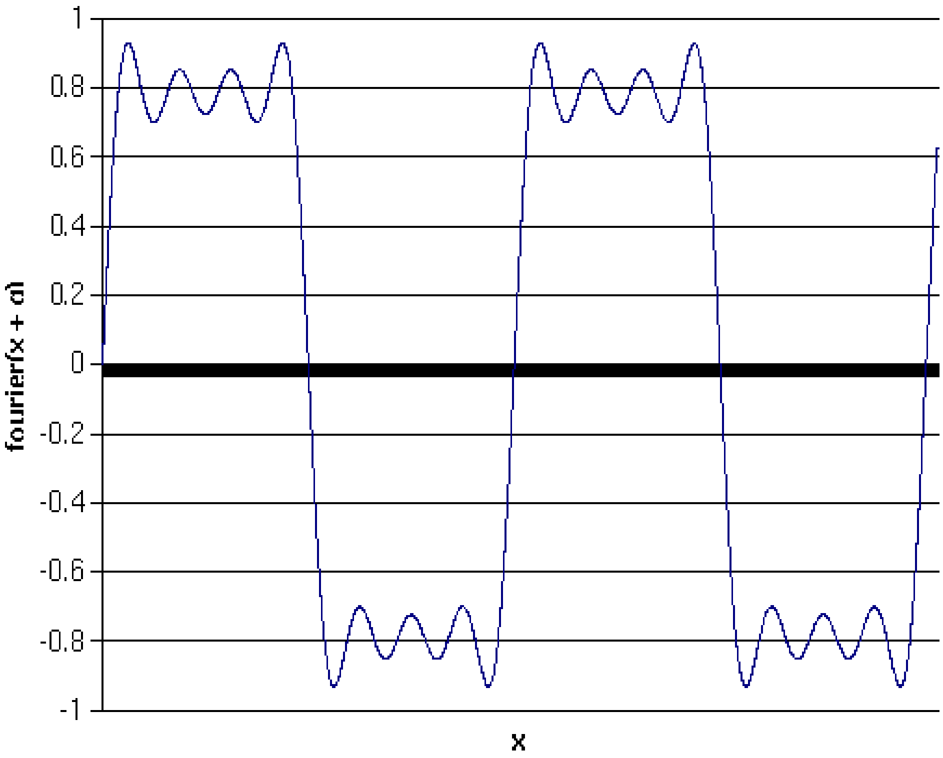


Figure 8: 4 term Fourier Series approximation

4 Animation - Sine Wave

The sine graph in the previous section can be given the appearance of 'travelling' by using an Excel macro. A macro is a piece of Visual Basic code that can update the values in the spreadsheet.

4.1 Setup

Setting this up was relatively simple. I copied the sheet from the previous exercise, keeping the names assigned to each cell / range of cells, and added in a $\delta\theta$ field (called dq) - note that q and dq actually correspond to cells B4 & B5, as this is necessary for the macro.

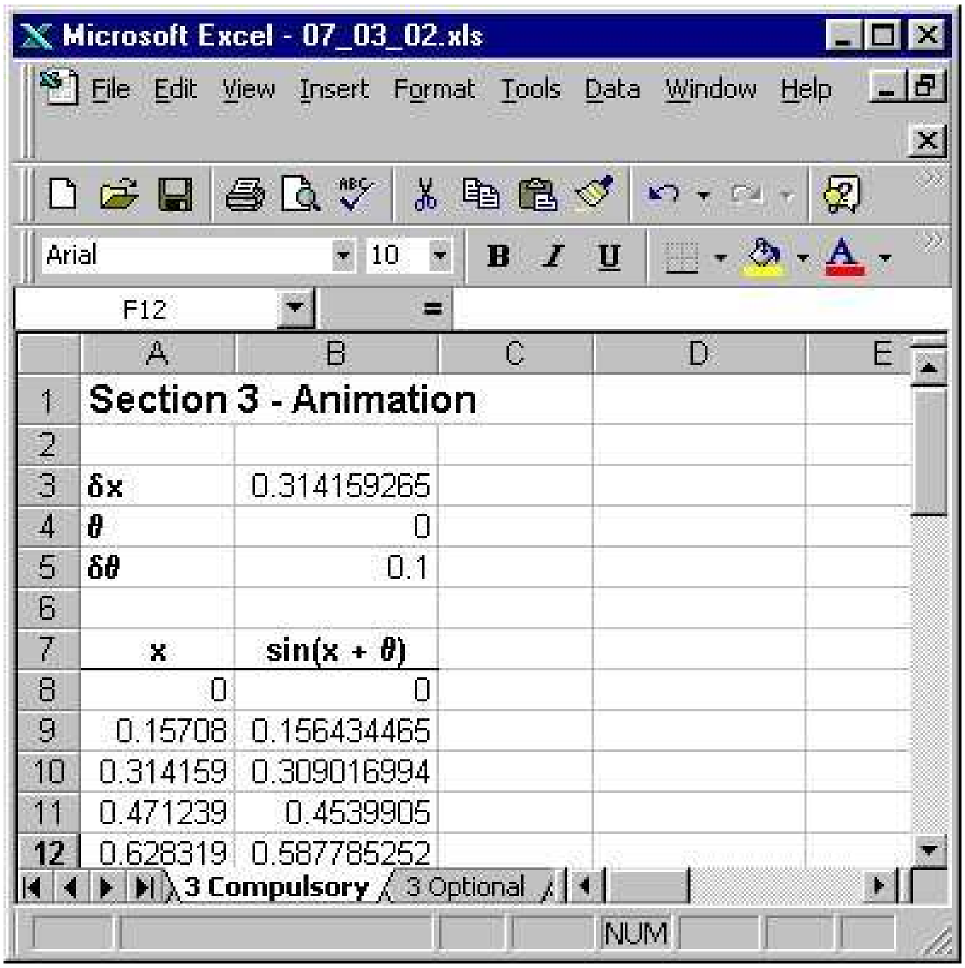


Figure 9: Initial Setup

4.2 Macro

The listing for the macro I used to update the phase variable is given below. This was assigned to a keyboard shortcut to make running it easier.

```
Sub inc_theta()

     [B4].Value = [B4].Value + [B5].Value

EndSub
```

That increments the value of θ by $\delta\theta$ once. For convenience, I changed the macro to loop 10 times, to produce a more animated effect.

```
Sub inc_theta()

  For i=1 To 10
      [B4].Value = [B4].Value + [B5].Value
  Next i

EndSub
```

4.3 Running the Macro

When you run the macro, it has the effect of incrementing the value of θ in small steps, and since Excel updates the graph each time you change the value of it's data, it has the effect of making the wave appear to travel from right to left. In figure 10 you can see several stages in the motion.

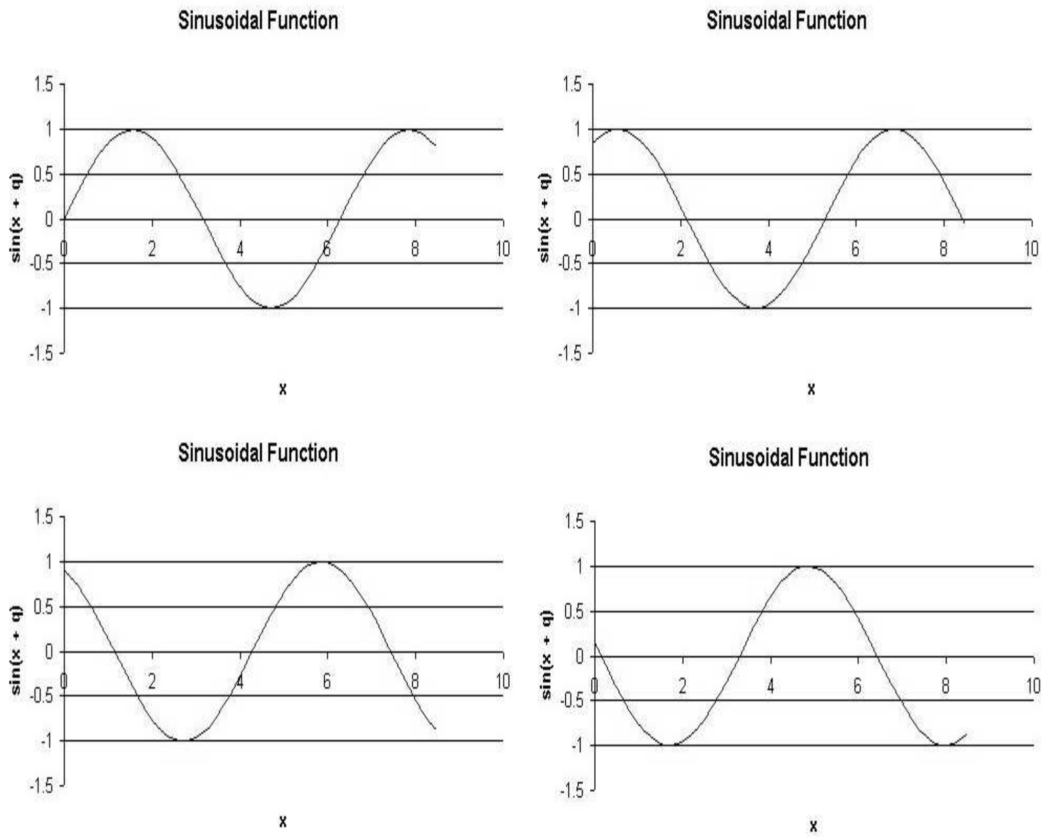


Figure 10: The moving wave, at +10 iterations per image

4.4 Fourier Series Animation

For the optional task in the animation section I took the sheet from the static Fourier example I did earlier, and adapted it to be animated. As in the previous example I added a field for $\delta\theta$, named dq. Since this was conveniently in the same place as $\delta\theta$ in the previous example, the macro I wrote there also worked in this example with no alteration. You can see the result of running that macro in figure 11.

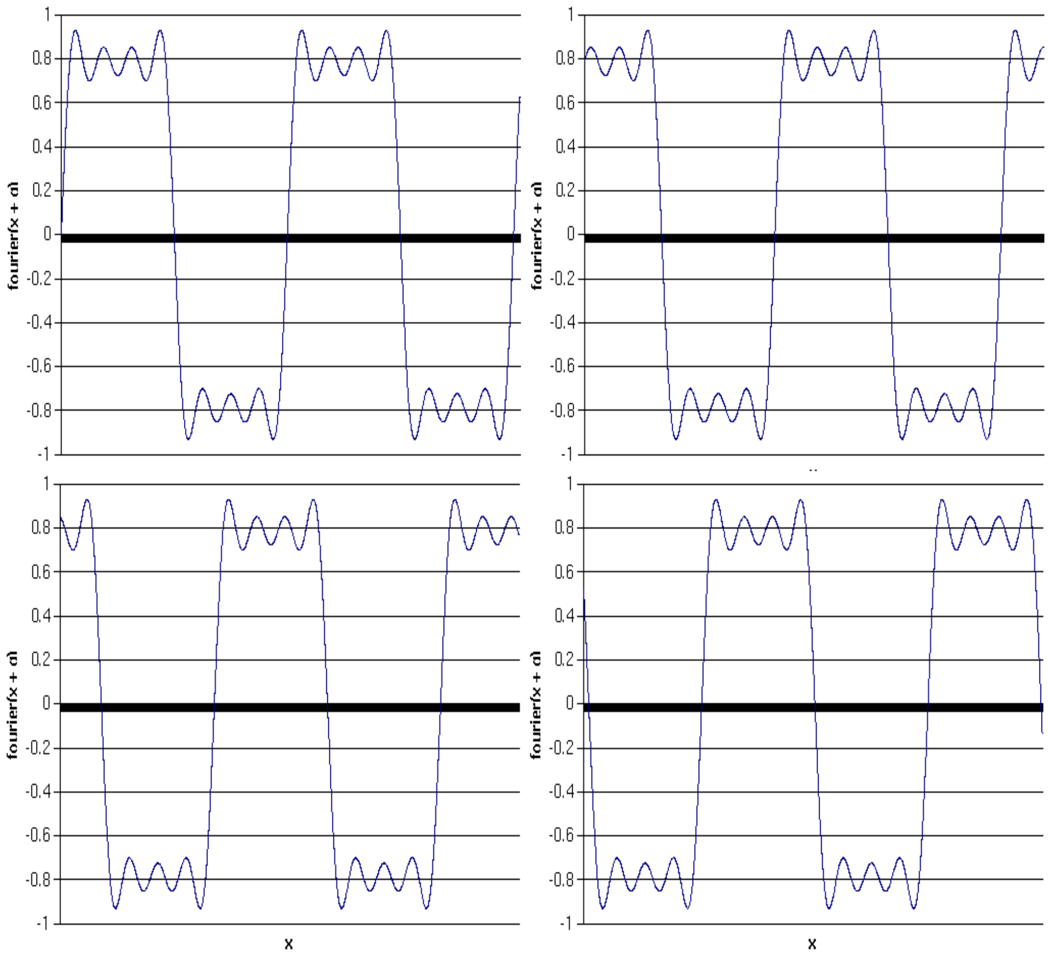


Figure 11: Animated Fourier series, at +10 iterations per image

5 Simulation - Simple Harmonic Oscillator

The object of this section was to try and model a system oscillating under SHM, without solving the equations abstractly. To do this, you can use the iterative features of Excel. The formulae for the position (X) and velocity (V) of a particle undergoing SHM are

$$\frac{dX}{dt} = V \tag{3}$$

$$\frac{dV}{dt} = -X \tag{4}$$

Since we will be dealing with discrete time intervals (albeit small ones) to approximate the SHM to something that Excel can solve, these can be reduced to difference equations. Given that the time between samples is T, then

$$X_{n+1} = X_n + V_nT \tag{5}$$

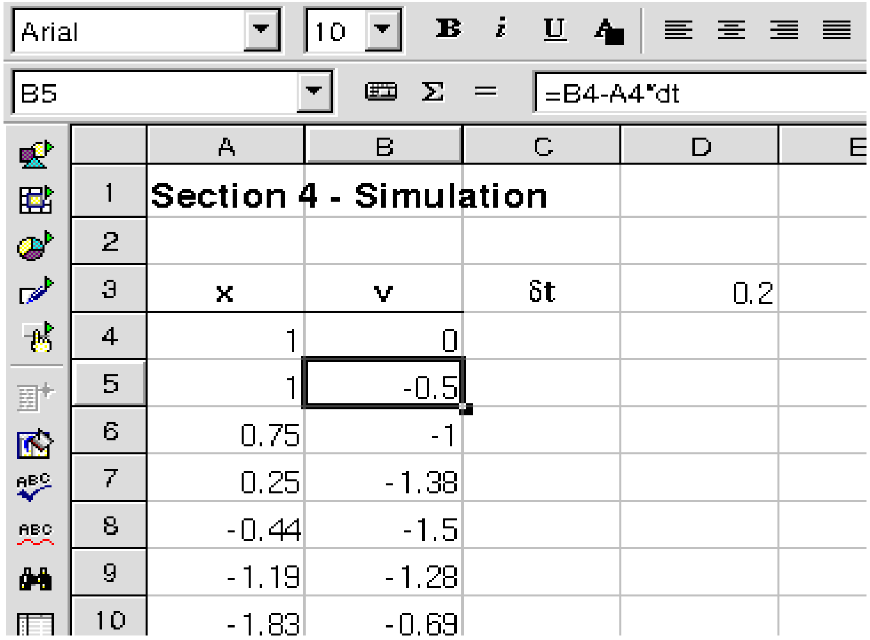
$$V_{n+1} = V_n - X_nT \tag{6}$$

(This is merely a rearrangement of the differential equations above). Note that this is an approximation, and as such will not be a completely accurate simulation, there are rounding errors, and other approximations, that are not present in the limit case. If we take a sufficiently small T, however, these errors are reasonably small, and unless a high degree of accuracy is required, it is an acceptable simulation.

5.1 First Attempt

The spreadsheet was setup with a time increment field (given the name *dt*) and 2 columns, one each for position and velocity. These two columns have the first cell with an initial value and were named X and V respectively, for the whole dataset. The rest of the columns were filled with formulae taken from equations 5 & 6. When translated into Excel notation, they become `=A4 + B4*dt` in the A (X) column and `=B4 - A4*dt` in the B (V) column. These formulae were filled down a hundred rows or so, to ensure we have enough iterations. Figure 12 shows the top few values with the default δt of 0.2. As you can see from the graph in figure 13, there is something very wrong with the oscillation.

The sinusoidal graph that this simulation is meant to produce is a constant function, rather than an increasing one we have produced here. That is, the maxima should all be the same height, rather than increasing as we have here. This appears to be an exponential increase - the maxima are about doubling for each iteration. This is what we would expect in a forced oscillation where we are at the resonant frequency, not what we want from the simple harmonic situation.



Arial | 10 | B5 | =B4-A4*dt

	A	B	C	D
1	Section 4 - Simulation			
2				
3	x	v	δt	0.2
4	1	0		
5	1	-0.5		
6	0.75	-1		
7	0.25	-1.38		
8	-0.44	-1.5		
9	-1.19	-1.28		
10	-1.83	-0.69		

Figure 12: Initial setup for simulation

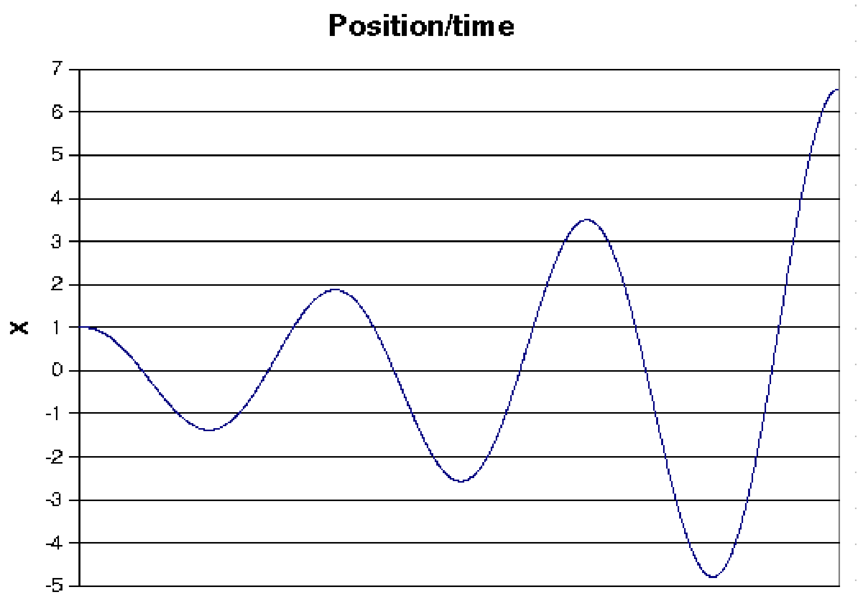


Figure 13: The (Incorrect) graph of the simulation

5.2 An Improved Simulation

We can try to improve the simulation by decreasing the time increment. That should give us a better simulation as we are approaching the limit of δt more closely. This seems to begin to have the desired effect, the maxima increase more slowly, but it also has the effect of reducing how many oscillations we are calculating (see figure 14). This is then not a good solution, as we can expect the simulation only to become good as $\delta t \to 0$.

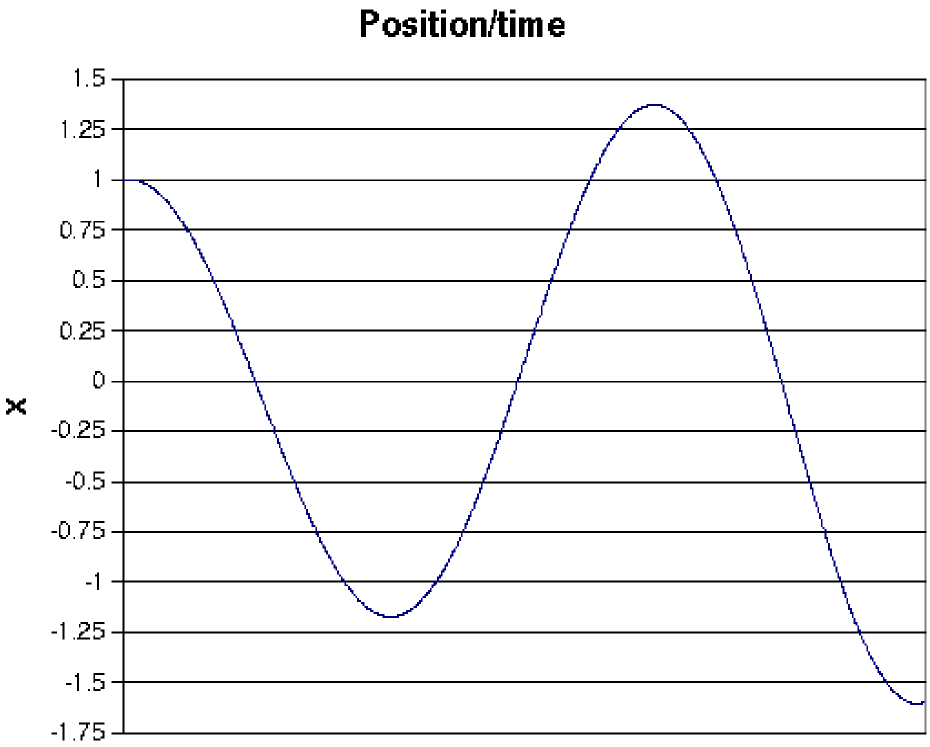


Figure 14: Graph with $\delta t = 0.1$

There is a better fix we can use to improve the accuracy of the integration. If we change the formula in one column, so that it uses the value in the same row from the other column, rather than the previous value, then that reduces the amount of error introduced. Equation 6 then becomes $V_{n+1} = V_n - X_{n+1}T$. As you can see from the resultant graph (figure 15) this is a much better approximation, and appears completely accurate within the viewable parameters set here. In fact even if you set $\delta t = 0.5$ the first and last maxima (7 complete oscillations) differ by only 0.004, so the deviation is very small.

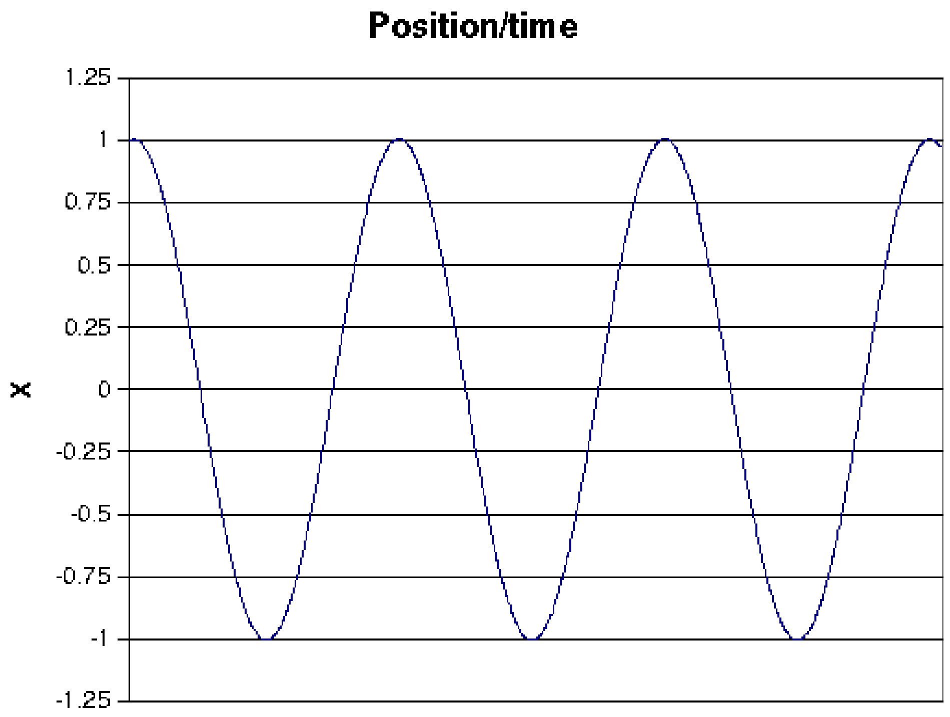


Figure 15: Graph with new formula and $\delta t = 0.2$

5.3 Damped Harmonic Oscillator

Starting from the sheet in the Simple Harmonic Oscillator section, I added in a variable λ, named l. This is the damping coefficient in the formula for velocity. Equations 4 & 6 then get the damping term added to them, as below, and we update the Excel formula to `=B4 - A5*dt - B4*l*dt`.

$$\frac{dV}{dt} = -X - \lambda V \tag{7}$$

$$V_{n+1} = V_n - X_n T - V_n \lambda \tag{8}$$

Having done that we can see the oscillation displayed in figure 16, and you can clearly see that it is in a light damping situation.

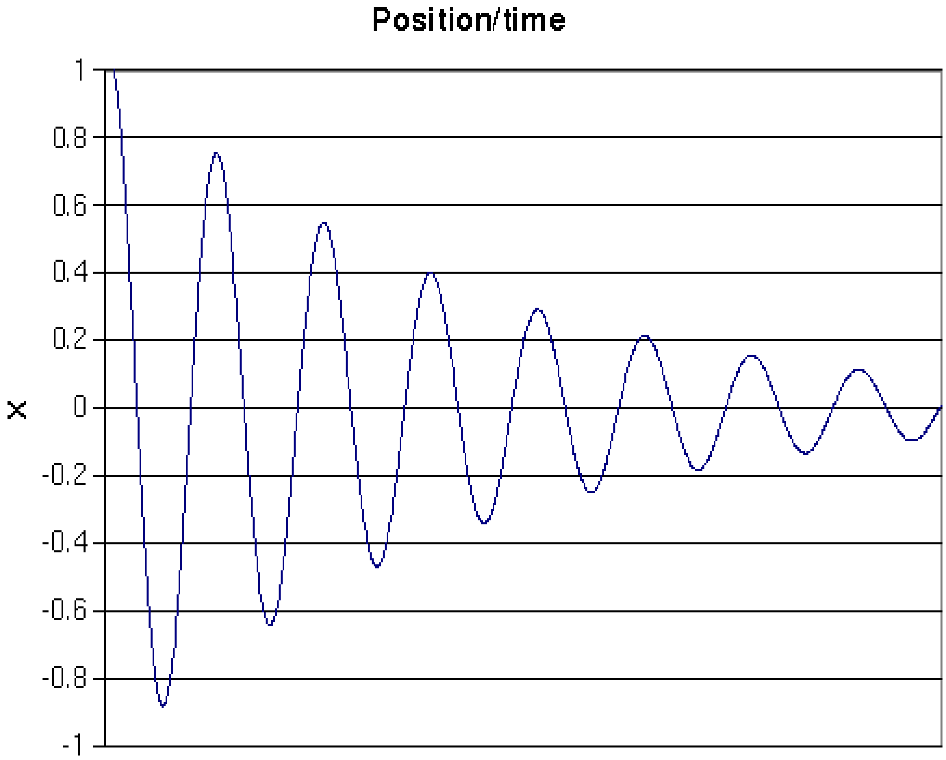


Figure 16: Light damping, $\lambda = 0.1$

To see how the oscillation varied with respect to λ, I wrote a macro to increase λ by 0.1, and watched the graph change. At about $\lambda = 1$ the motion changed from light damping to critical damping, where less than a full oscillation is carried out before the motion stops completely. If this was a simulation to model, for example, car suspension, that is the point that you would want to aim for, so that there is the maximum amount of give in the springs, but it doesn't oscillate as the car goes over bumps. That problem also includes a forcing term though, and the equation has the form

$$m\frac{d^2x}{dt^2} + a\frac{dx}{dt} + kx = F(x)$$

You can see the graphs for several values of λ in figure 17

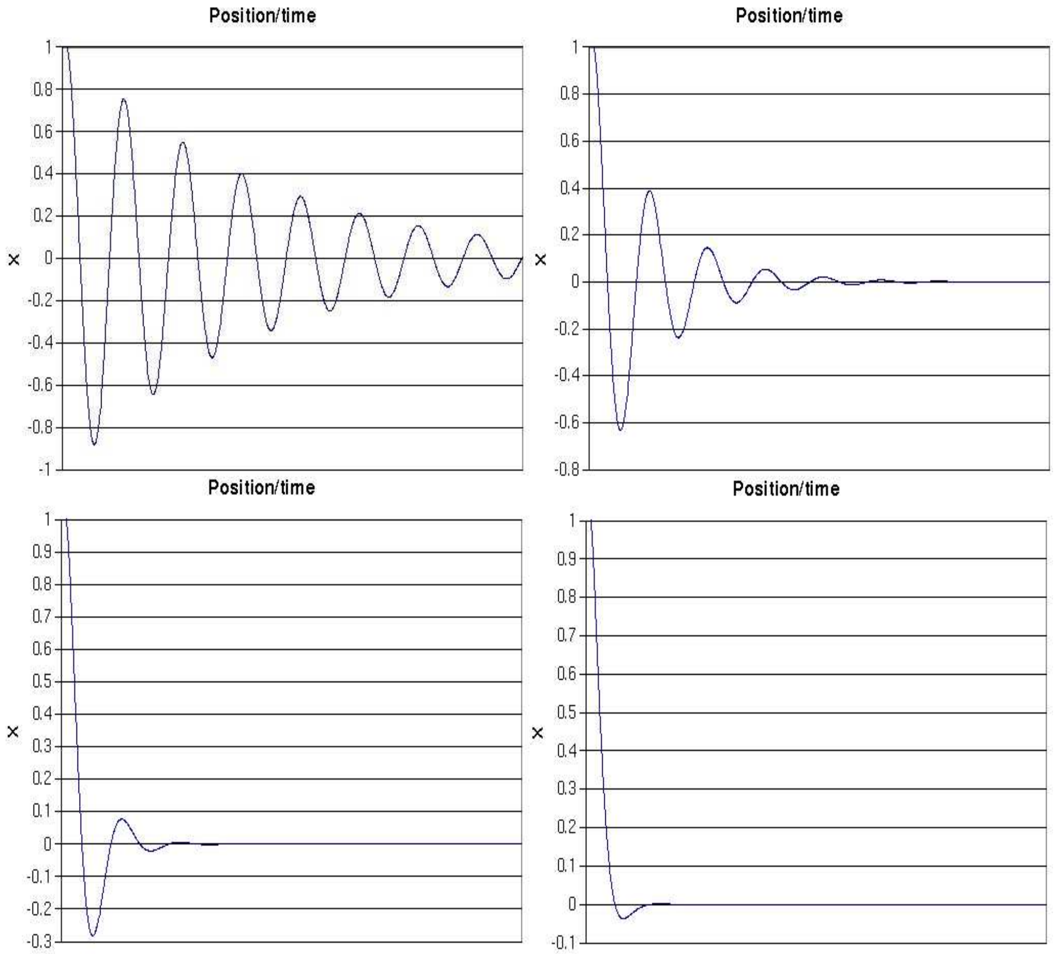


Figure 17: In order, $\lambda = 0.1$, $\lambda = 0.3$, $\lambda = 0.7$, $\lambda = 1.2$

6 Conclusion

There are a lot of things you can do with a spreadsheet that can aid with researching physical situations. In the most basic form, you can use them for collating data and performing routine calculations on that data. This is probably what most people use them for most of the time, and it works well, because they will invariably be more accurate than a human doing the same job. You do have to be aware, however, that this can give the appearance of greater accuracy than is actually present, since there will be errors an inaccuracies in the data collection methods, and the spreadsheet will tend to display a large number of decimal places, of which often only the first few are relevant, because of the errors elsewhere in the system.

The other main use for spreadsheets is in simulations of physical systems, where you can test out mathematical models for systems, and compare them against measured results. There are more pitfalls to be avoided here, since the simulations can take what look like good equations, and produce incorrect results, as we saw in the simple harmonic motion example above. This was not a problem in that particular example, because we knew what to expect, and hence that is was wrong. This is not always the case, and so if the results are wrong, it may not be a case of incorrect formulae, but of the implementation introducing errors.

Using a spreadsheet can definitely help you in a lot of situations, but it is always worth remembering its limitations, and taking them into account when considering results from them.