Damped Simple Harmonic Oscillator

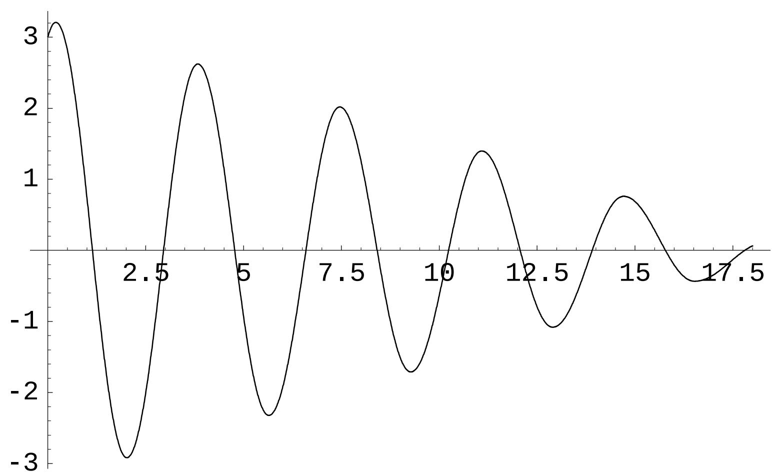
- **Read WELL before typing ANYTHING!**
  1. Below there are two sections: one labled *Mathematica*, and the other Maple. They represent the same programme written once in *Mathematica* and once in Maple. Look at each of these sections and compare the programmes.
  2. The programmes appear in the boxes below the word "INPUT". Only what's IN THE BOX is what you are required to type on your computer. DO NOT type anything else.
  3. The part called OUTPUT is what you are expected to see on your screen.
  4. The place where the lines end on this paper is not necessarily the place where the line that you type should end, so DO NOT add spaces to produce a programme exactly looking like the one here. In *Mathematica* after you finish each line press "ENTER" this will take you to the next line but will not EXECUTE the programme. Only when you finish typing press SHIFT+ENTER or the ENTER in numpad to execute.
  In Maple press SHIFT+ENTER to go from one line to the next, and EXECUTE using ENTER. Notice that *Mathematica* and Maple use these key combinations oppositely!
  5. Before starting SAVE whatever work you have previously written and type in *Mathematica* Exit and in Maple type quit;
  6. Notice that in Maple you MUST type a semi-colon ";" after each statement if you want its output to be displayed on the screen and type a colon ":" after the statement if you don't wish to see the output, it will still be in the memory but not on the screen. In *Mathematica* the semi-colon means DO NOT type the output (opposite to Maple), but the colon has other function that will not be dealt with now.
  7. DO NOT change the CASE of the text, this means when you see a capital or small lettre (upper or lower case) type it as it appears here. The computer will not understand the word with changed case.

Mathematica:

- **INPUT:**

```
t = 0; A = 3; B = 0.5; c = 1; m = 1; aa = {}; x0 = 3;
v0 = 2; y1 = x0; y2 = v0; t1 = 18; h = .005;
While[t < t1,
 f1 = y2;
 f2 = -A / m * y1 - B / m * Sign[y2];
 If[
  y2 * (y2 + h * f2) > 0 || Abs[y1] > c / A,
  y1 = y1 + h * f1;
  y2 = y2 + h * f2;
 ];
 t = t + h;
 AppendTo[aa, {t, y1}];
]
ListPlot[aa, PlotJoined → True, PlotRange → All];
```

- **OUTPUT:**



 Instructor: Prof. Dia-Eddin Arafah, 2001-2002

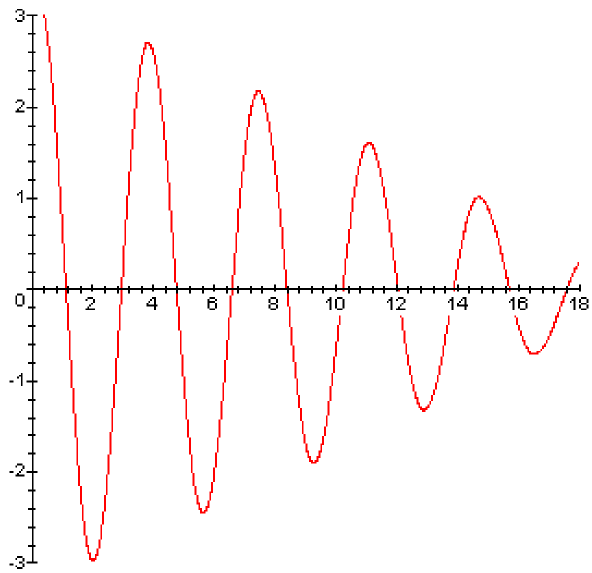
Maple:

- **INPUT:**

```
t0 := 0; A := 3; B := 0.5; c := 1; m := 1; x0 := 3; v0 := 2;
y1 := x0; y2 := v0; t1 := 18; h := .01; dat := [t0, x0] :
  for t from t0 by h while t < t1 do
f1 := y2;
f2 := -A / m * y1 - B / m * sign (y2);
if (y2 * (y2 + h * f2) > 0 or abs (y1) > c / A) then
y1 := y1 + h * f1;
y2 := y2 + h * f2;
fi;
dat := (dat, [t, y1]);
od : dat := [dat] : plot (dat, 0..t1, -3. .3);
```

- **OUTPUT:**

```
t0 := 0
 A := 3
B := .5
 c := 1
 m := 1
x0 := 3
v0 := 2
y1 := 3
y2 := 2
t1 := 18
h := .01
```



Exercises and lab work:

- **Explain why does the amplitude of oscillation increase (and not decrease) when the step h is increased beyond 0.03. Suggest how one can prevent this from happening but still use this step size, mention the idea with no details.**

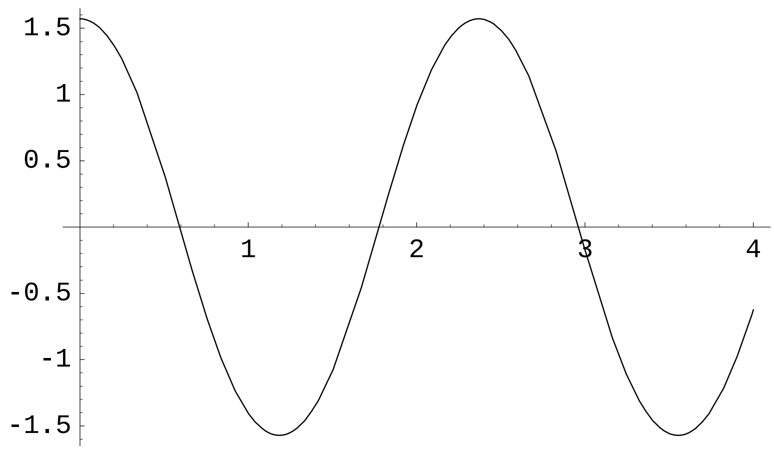
- **Set up the Lagrangian and then derive the equation of motion for the simple pendulum.With no small angle approximation, the equation should read: $\ddot{\theta} + \omega_0^2 \sin\theta = 0$. Assuming the length of the pendulum is 1m and that the bob starts motion from rest, we want to study the motion once by using the small-angle angle approximation and once without it. The first calculation shows the solution of the equation without approximation assuming that the initial angle is $\pi/2$. The second uses the small-angle approximation for the same initial conditions. The third plot shows the two solutions on the same graph.**

- **Calculate the period of motion for the pendulum for the two cases. (Hint: use *Mathematica*'s command FindRoot)**

- **What do you think is the difference between DSolve and NDSolve, and why were they used in different places?**

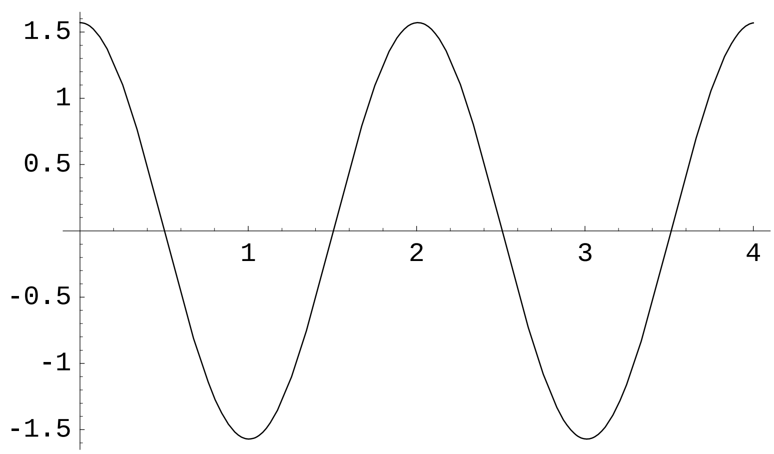
```
NoApproximation = NDSolve[
   {y''[x] + 9.8 Sin[y[x]] == 0, y[0] == π/2, y'[0] == 0}, y, {x, 0, 6}][[1]]
Plot[y[x] /. NoApproximation, {x, 0, 4}];
```

```
{y → InterpolatingFunction[{{0., 6.}}, <>]}
```



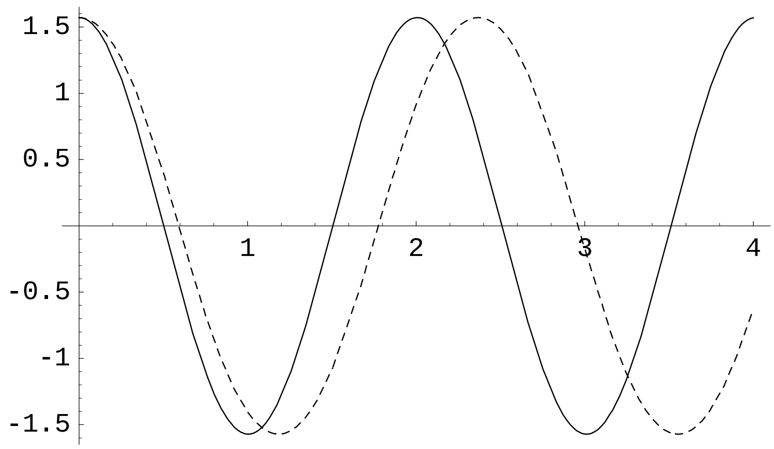

```
WithApproximation =
 DSolve[{y''[x] + 9.8 y[x] == 0, y[0] == π/2, y'[0] == 0}, y, x][[1]]
Plot[y[x] /. WithApproximation, {x, 0, 4}];
```

```
{y → Function[{x}, 1.5708 Cos[3.1305 x] + 0. Sin[3.1305 x]]}
```



Instructor: Prof. Dia-Eddin Arafah, 2001-2002

```
Plot[{y[x] /. WithApproximation, y[x] /. NoApproximation},
  {x, 0, 4}, PlotStyle → {Dashing[{1, 0}], Dashing[{0.01, 0.01}]}];
```



- **The following programme finds the plot of the motion without approximation but in Maple. Execute it and then compare the value of the NoApproximation at the first root. What do you conclude?**

```
n := 40;
ff := dsolve
   ({D (D (y)) (x) + (9.8 / 1) * sin (y (x)) = 0, y (0) = Pi / 2, D (y) (0) = 0},
    y (x), type = numeric, output = listprocedure);
fx := subs (ff, y (x));
X := seq (i / 10., i = 0..n) :
    Y := [seq (fx (i), i = X)] :
     dat := [seq ([X[i], Y[i]], i = 1..nops ([X]))] :
     plot (dat);
```