Numerical Solution of Driven Mass Spring Damper Systems

Jason Underdown

Instructions

Run this worksheet by pressing the "!!!" button on the toolbar above. Once it has completed running, (it may take a minute or so to run depending on your computer) scroll down to the bottom and click on the graph. This will put the animation toolbar at the top of this document window. Press the play button.

Some Useful Input Functions

Below we define some functions which allow us to create interesting forcing functions (inputs). In what follows, \( T \) = period and \( \omega \) = the angular frequency of the forcing or driving function.

\[
\begin{align*}
&\text{SawTooth} := t \rightarrow 2t - 2 \cdot \text{floor}(t) - 1 : \\
&T := \frac{2 \pi}{\omega} : \\
&\text{TriangularWave} := t \rightarrow \text{abs}\left(2 \cdot \text{SawTooth}\left(\frac{(2t - 1)}{4}\right)\right) - 1 : \\
&\text{SquareWave} := t \rightarrow \text{signum}(\text{TriangularWave}(t)) : \\
&\text{Switch} := t \rightarrow 1 - \text{Heaviside}(t - T) : \\
&\text{DoubleSwitch} := t \rightarrow 1 - \text{Heaviside}(t - T) + \text{Heaviside}(t - 2T) - \text{Heaviside}(t - 3T) : \\
&\text{DoubleFlipSwitch} := t \rightarrow 1 - 2 \cdot \text{Heaviside}(t - 2T) + 2 \cdot \text{Heaviside}(t - 4T) - \text{Heaviside}(t - 6T) : \\
&\text{Boxcar} := t \rightarrow \text{Heaviside}(t - T) - \text{Heaviside}(t - 2T) : \\
&\text{DoubleBoxcar} := t \rightarrow \text{Heaviside}(t - T) - \text{Heaviside}(t - 2T) + \text{Heaviside}(t - 3T) - \text{Heaviside}(t - 4T) :
\end{align*}
\]

Setup and Main Solution Loop

The code below simulates a mass, spring, dashpot/damper system governed by the equation:

\[
mx''(t) + cx'(t) + kx(t) = f(t).
\]

Once you define a forcing function (the input), the loop below solves the differential equation (numerically) for \( \omega \) values which vary from \( h \) (the step size) to \( Nh \). The loop plots the forcing function \( f(t) \) as well as the solution (output) \( x(t) \) on the same set of axes. Finally, we use the display function from the plots library to generate an animated sequence from all of the plots generated in the loop. The
animation shows how $x(t)$ varies with the angular frequency of the forcing function, $\omega$.

Feel free to experiment with the code and change any of the values defined before the loop. Depending on your computer's available memory, you probably don't want to set $N$ any higher than about 500.

```maple
> m := 1 : c := 0 : k := 25 : Amp := 1 :
> solve(m*r^2 + c*r + k = 0, r) # characteristic equation
> f := t->Amp*SquareWave(2*t/T):
> #f:= t->Amp*cos(omega*t):
> #f:= t->Amp*DoubleFlipSwitch(t):
> IVs := x(0) = 0, x'(0) = 0 : # initial values
> tL := 30 : # t plot limit
> xL := 1.1*Amp : # x plot limit
> N := 100 : # number of frames in the animation
> h := .1 : # step size for incrementing omega in the following loop
> for i from 1 to N do
    omega := i*h:
    pin := plot(f(t), t = 0 ..tL, y=-xL..xL, color = navy) : # plot of forcing function f(t)
    DE := m*x''(t) + c*x'(t) + k*x(t) = f(t):
    out := dsolve({DE, IVs}, x(t), type = numeric):
    pout := plots[odeplot](out, [t, x(t)], t = 0 ..tL, numpoints = 600, view = [0..tL, -xL..xL],
                       caption = typeset("w= %1", omega)) :
    L[i] := plots[display]([pin, pout]) : # combine plot of f(t) and x(t) into one plot
end do:
> plots[display](seq(L[i], i = 1 ..N), insequence = true)
```