Week #8 : Second Order Differential Equations

Goals:

- Express real world situations in terms of second order linear differential equations.
- Describe the difference between homogeneous and nonhomogeneous second order linear differential equations.
- Use MATLAB to solve linear and nonlinear second order differential equations, both homogeneous and nonhomogeneous.

Generating Numerical Solutions with MATLAB

Problem. Search for "first order ODEs" or "initial value problem" in MATLAB help.

Problem. What form of differential equation does MATLAB assume we have?

Note that differential equation solvers are a big component of MATLAB: you may have to do some reading to sort out the kind of equation you have, and what the appropriate solving tool is.

Problem. Which of the solvers is recommended as a “first try” solver?

ode45

The first solver to reach for in MATLAB is `ode45`. To run it, we need

- the DE function **in form** $\frac{dy}{dt} = f(t, y)$;

- the time span for the solution/simulation, $[t_0, t_{\text{end}}]$; and

- the initial condition $(y(t_0))$

Temperature Model - Newton's Law of Heating and Cooling

The temperature of an object, y, changes at a rate proportional to the temperature difference between object, y, and its environment, T_{ext}.

Problem. Translate this law into a mathematical statement.

$$\frac{dy}{dt} = -k(y - T_{\text{ext}})$$

Solution Interpretation: a **solution** to a differential equation is the same as a **prediction**.

```
% Define temp DE in the form dy/dt = f(t, y)
% and set other constants
k = 0.7;  % /min

T_ext = 20; % external/environment temp

DE = @(t, y)   -k (y - T_ext);
```

Example - Step 2

```
% Solve based on initial condition
y0 = 100;  % initial temperature

tspan = [0, 30];  % interval for solution

[t, y] = ode45(DE, tspan_, y0);

plot(t, y);
```

To decipher and work with the output, it is critical that you understand what MATLAB provides.

- The final values of `t` and `y` are

- `t` starts at

- `t` ends at

- `y` starts at

Basic Process

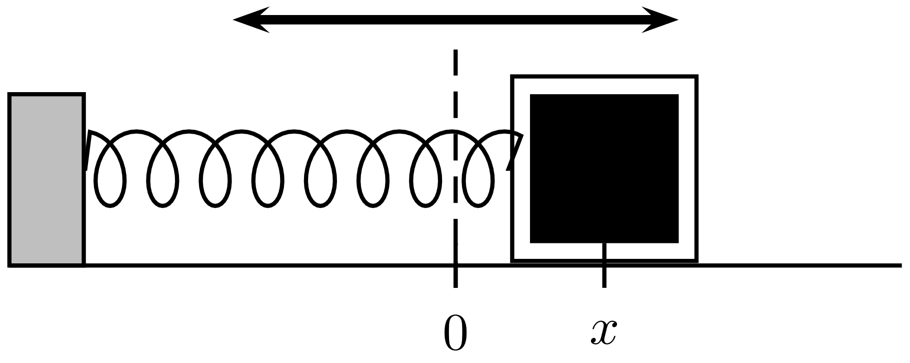
How does MATLAB do it?

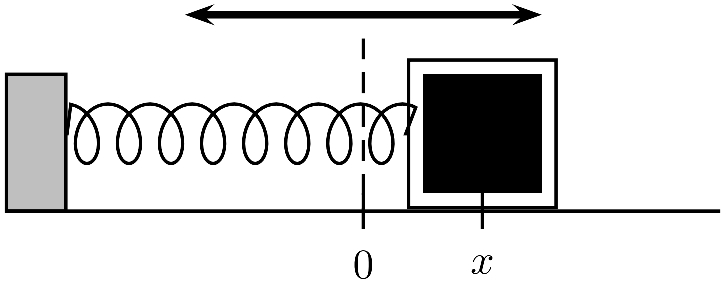
Second-Order Linear Equations - Spring System

So far we have seen examples of **first-order DEs**, or equations with first derivatives of some unknown function.

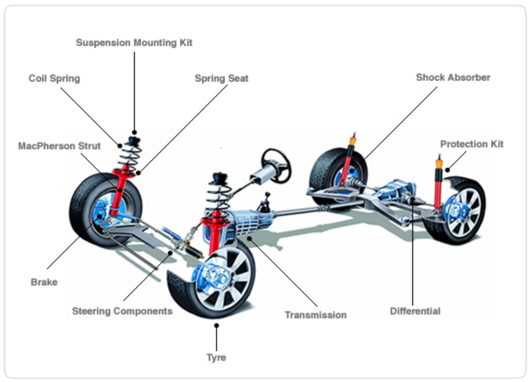
In the following examples, we will expand our study to differential equations with **second or higher derivatives**.

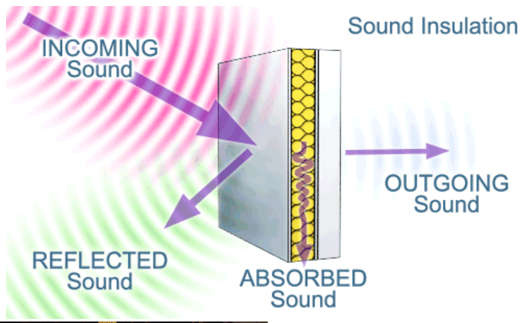
One classic source of differential equations of this type comes from analyzing the forces on a block at the end of a spring.

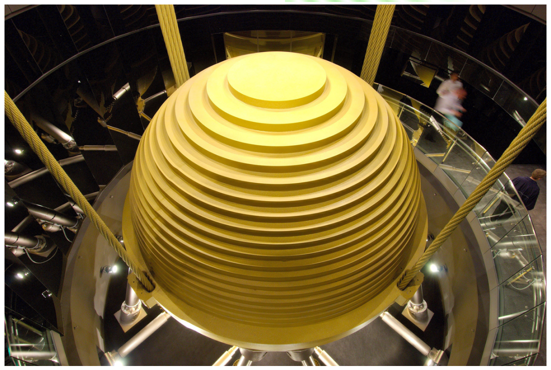




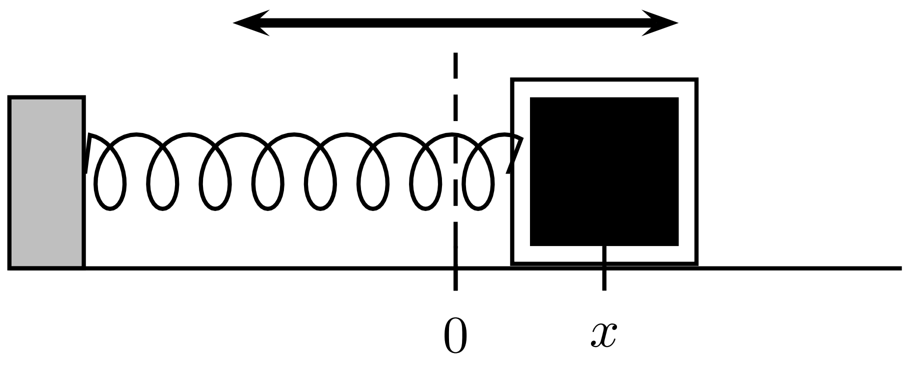
While the mathematics behind this simple system will be very interesting in their own right, we should also note at the outset that the simple spring/mass model can be applied to a wide variety of not-so-obviously related real-world problems.







Spring System Analysis



In this system, how would you describe x in words?

Problem. Draw a free-body diagram for the mass. Indicate the magnitude of the forces, assuming

- the mass of the block is m kg, and
- the spring constant (in N/m) is given by the constant k.

Let us work with our intuition about this system before beginning the mathematics.

Problem. If the spring is very stiff, is k large or small?

Period: the length of time to complete one full cycle/oscillation.

Problem. If we increase the stiffness of the spring, do you expect the *period* of the oscillations to increase or decrease? Why?

If we increase the mass, do you expect the *period* of the oscillations to increase or decrease? Why?

Problem. If we know k and m, and assume that friction is negligible, should we be able to determine the exact period of the oscillations?

From the work so far, can we easily find the formula for the period?

The spring system is an excellent introduction to higher-order differential equations because

- we all have an intuition about how it *should* work physically,
- the mathematics and physics are simple, and
- there's no obvious way to predict critical features (e.g. the period) from the given information.

We clearly need some new tools!

Spring System as a DE

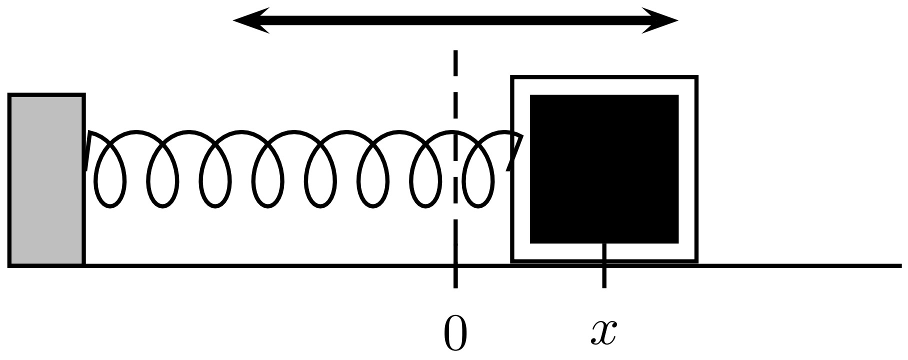
Problem. Use Newton's second law, $F = ma$, to construct an equation involving the position $x(t)$.

What order of differential equation does $F = ma$ produce for this spring/mass system?

To simplify matters temporarily, let us assume that both $k = 1$ N/m and $m = 1$ kg.
Problem. Rewrite the previous differential equation.

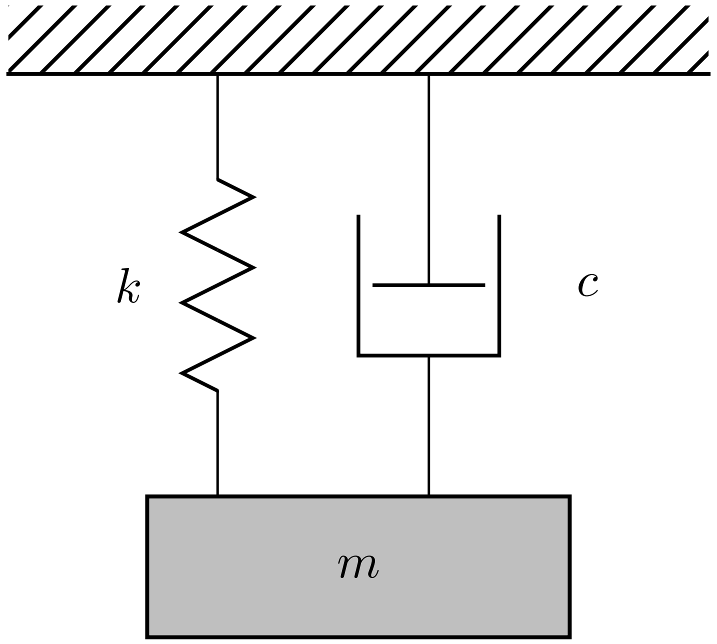
This differential equation invites us to find a function $x(t)$ whose second derivative is its own negative. What function(s) would satisfy that?

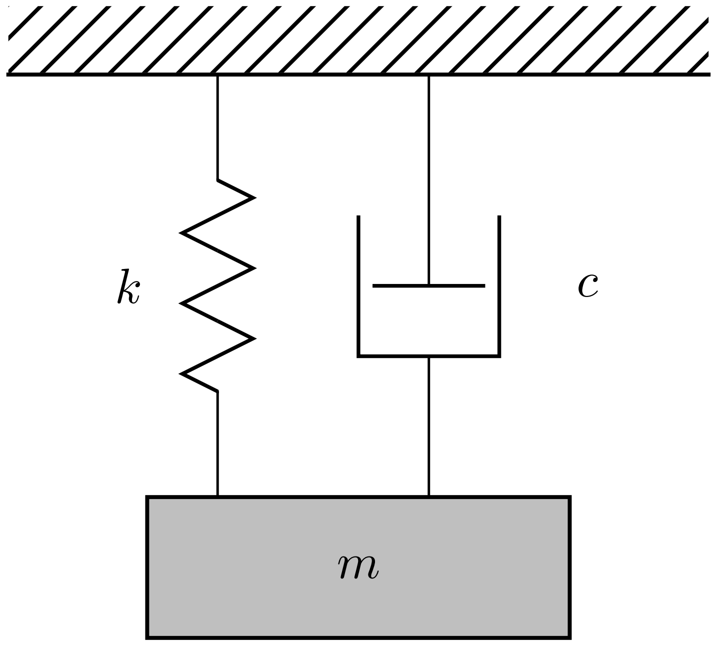
Problem. Having found two (and more) solutions to the differential equation for the spring/mass system, how does this family of solutions map back to the spring system?



Mechanical Vibrations - Spring-mass system

We now consider an extension to the spring/mass system seen earlier
Consider a mass m hanging on the end of a vertical spring; the spring has a natural length corresponding to $x = 0$.
We have now added a damper, which exerts a force proportional to the velocity of the oscillating mass.
Problem. Using Newton's second law, build a differential equation that governs the system.




k
c
m

Problem. Consider a mass of 0.5 kg with spring constant $k = 2\ \mathrm{N} \cdot \mathrm{m}^{-1}$ in an undamped unforced system. Assume the mass is displaced 0.4 m from equilibrium and released. Describe the long-term behaviour of the system.

Converting Higher-Order DEs to 1st Order Systems

To solve a higher-order DE using MATLAB, we need to convert it first into a larger **first-order system** of differential equations.

Notation: In this section,

- vectors with be written vector hats, e.g. $\vec{x}$,
- elements of a vector will be noted with subscripts, e.g. w_1, y_2, and
- other scalars will be in lower-case, e.g. c, λ.

$$x'' = -cx' - kx$$

Problem. For the spring/mass DE, define a new vector of 2 variables, $\vec{w}$, that will allow the conversion of the second-order system to a first-order system.

$$x'' = -cx' - kx$$

Problem. Define the derivative of $\vec{w}$ in terms of $\vec{w}$ itself, making use of the DE as necessary.

This is now a **first-order system** of differential equations.
The variable we are most interested in for this example is $w_1 = x$, the *position* of the mass.

MATLAB Function Files

Our differential equation is now sufficiently complicated that writing it out in one line is difficult. We will need another MATLAB tool to proceed: a MATLAB function file.

Syntax A MATLAB function file

- has as its first line the form
  `function ret = f(....)`
- `ref` is the **return variable** and must be defined before the end of the function;
- `f` should be the same as the filename, with the `.m` extension;
- the `....` can be any list of input variables.

Problem. Define a new **MATLAB function `.m` file** called **`hypotenuse`** that takes in two lengths `a` and `b`, and returns the length of the hypotenuse for a right-angle triangle with side lengths a and b.

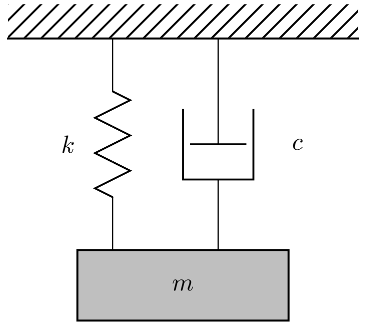
Problem. Define a new **MATLAB function** `.m` **file** called `springDE` which computes the derivative of $\frac{d\vec{w}}{dt}$.

$$\frac{dw_1}{dt} = w_2$$
$$\frac{dw_2}{dt} = -kw_1 - cw_2$$

Problem. Write a MATLAB script that graphs the solution (i.e. predicted motion) for the system described earlier:
A mass of 0.5 kg with spring constant $k = 2 \text{ N} \cdot \text{m}^{-1}$ in an undamped unforced system. Assume the mass is displaced 0.4 m from equilibrium and released.

Unforced Spring/Mass System - Patterns of Behaviour

Using MATLAB, we can now easily study the impact of different levels of damping on the behaviour of a spring/mass system.

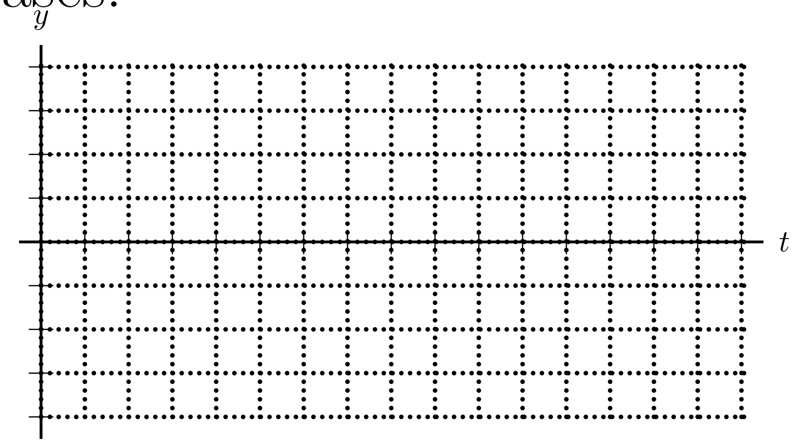


A famous result in physics is that when the damping coefficient

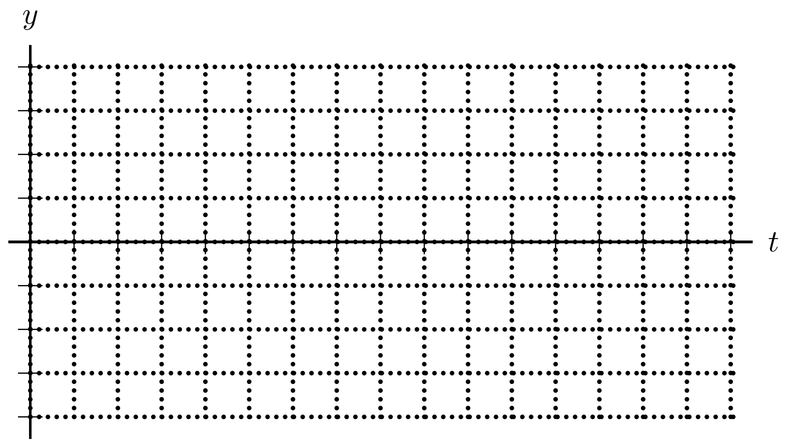
- c is 0, the system will be **undamped**;
- c is **less** than $\sqrt{4km}$, the system will be **under-damped**;
- c is **equal** to $\sqrt{4km}$, the system will be **critically damped**;
- c is **greater** than $\sqrt{4km}$, the system will be **over-damped**;

Problem. Using a spring constant of $k = 25$ N/m and mass $m = 1$ kg, find the critical damping level.

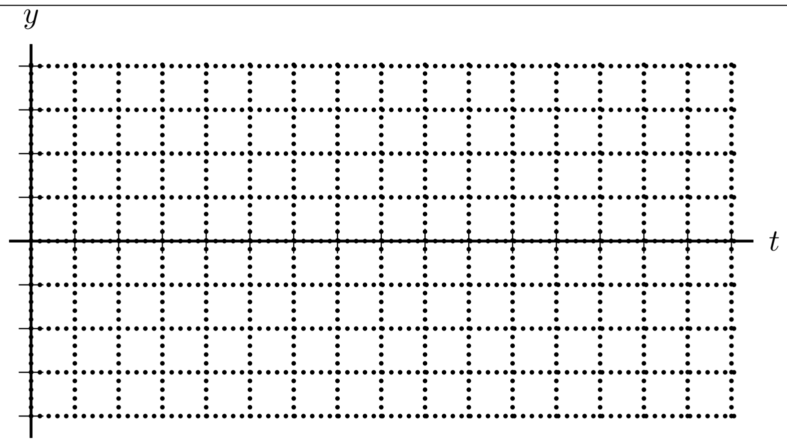
Problem. Use MATLAB to obtain graphs for all four spring/mass cases.



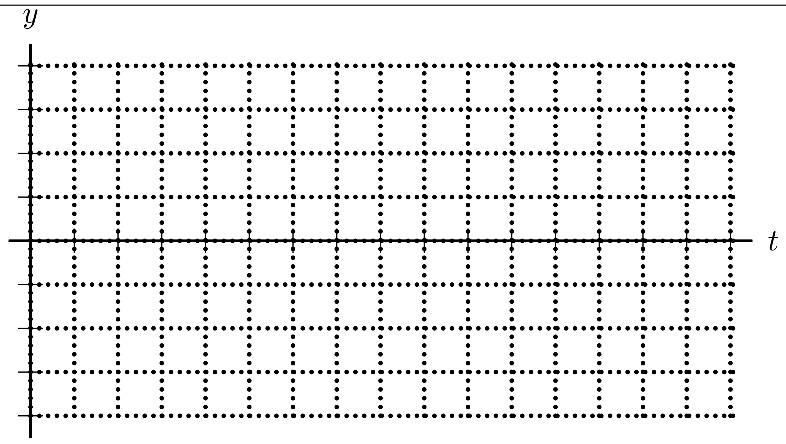
Undamped



Under-Damped



Critically Damped



Over-Damped

Nonhomogeneous Linear Differential Equations

The general form **homogeneous**:

$$y^{(n)} + a_{n-1}y^{(n-1)} + a_{n-2}y^{(n-2)} + \ldots + a_1y' + a_0y = 0$$

This week, we explore how to deal with equations with more interesting right-hand sides.

The general solution of a nonhomogeneous n-th order linear differential equation has the form

$$y = \underbrace{c_1y_1 + \cdots + c_ny_n}_{y_c} + y_{NH},$$

where

- $y_1, \ldots, y_n$ span the solution space to the corresponding **homogeneous** equation,
- $c_1, \ldots, c_n$ are arbitrary constants,
- y_c, the collection of $y_1, \ldots y_n$ and $c_1, \ldots c_N$, is called the "complementary solution" to the **homogeneous** equation, and
- y_{NH} is a specific solution to the **non**homogeneous equation.

Basic Strategy for Nonhomogeneous Linear Equations

1. Find a basis for the solution space of the **corresponding homogeneous equation**. Call those solutions $y_1, \ldots, y_n$.

2. Find a **single** solution, y_{NH}, to the **nonhomogeneous** equation.

3. The general solution to the nonhomogeneous DE will be

$$y = y_{NH} + \underbrace{(c_1 y_1 + \ldots c_n y_n)}_{y_c}$$

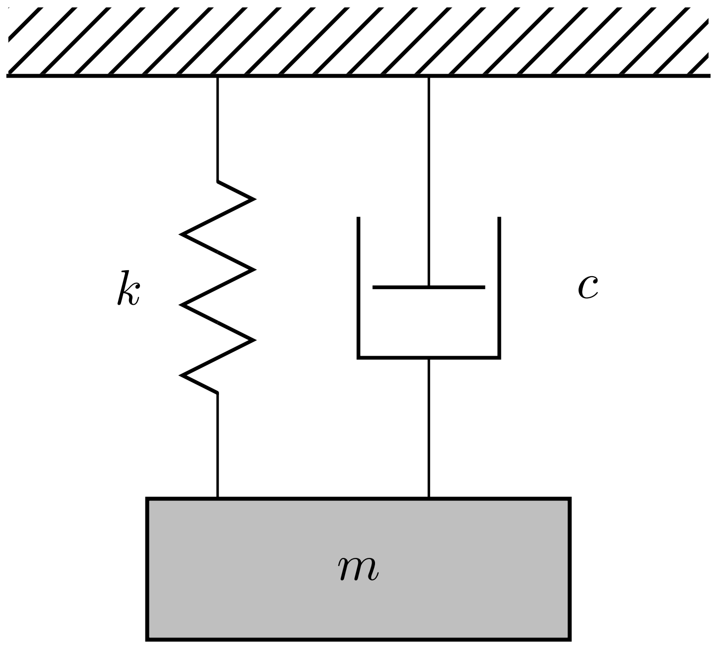
Periodic External Forces - Applications

We will next explore some of the ramifications of the nonhomogeneous solution form

$$y = \underbrace{c_1 y_1 + \cdots + c_n y_n}_{y_c} + y_{NH}$$

in several application areas.

Spring/Mass System



Problem. Write out the DE for the position of the mass, given $F_{\text{ext}} = A\sin(Bt)$.

We will explore the solutions of/predictions for the spring/mass system (and RLC circuits!) through changing parameters in a simulation. We will consider both

- graphs of the solutions, and
- mathematical forms of the solutions.

Your job during the demonstrations is to track:

- the amount of friction,
- the *natural* frequency (in y_c), and
- the *stimulating* frequency (in Fe and y_{NH})

Unforced Motion

Setting $F_{\text{ext}} = 0$, review the effect of increasing the friction coefficient c on the y_c solution.

What is the y_{NH} solution for all these simulations, and why?

Forced, but Undamped Motion

Now we turn off the friction (by setting $c = 0$) to zero, and start to apply the external force.
We set $F_{\text{ext}} = 10 \sin(1.0t)$.
What is the form of y_c, and what is the form of y_{NH}?

Experiment with the frequency in F_{ext}, in the range 0.5 to 3, or well above 5. Does the motion of the mass over time make sense?

Resonance

Set $F_{\text{ext}} = 1\sin(5t)$. What is special about the frequency 5 rad/s?

Describe the graph of the solution.

Give the mathematical reasons for the graph of the simulation.

The condition of having the external force at *exactly* the same frequency as the natural vibrations is called **resonance.** Note: for true resonance, there must be *no friction.*

Near-Resonance

Admitedly, resonance is a unique event requiring **perfect** matching of the stimulating force with the natural frequency.

Explore frequencies in F_{ext} *close to* 5. Describe the resulting solutions, both for their amplitude and any other response.

Give the mathematical reasons for the shape of the graph for near-resonance response.

Note again that there is *no friction* with this response.

Adding in Friction - Practical Resonance

Set $F_{\text{ext}} = 1\sin(5t)$ again, but set friction c to 0.1 Compare this solution to the one without friction, both graphically and mathematically.

Gradually increase the amount of friction. Describe how the solution changes.

How is this more realistic than the undamped case?

How do the mathematical forms of the pure-resonance (undamped) and practical resonance (some damping) compare?

Transient and Steady-State Solutions

More generally, when there is friction in the spring/mass system and an oscillatory external force, we can break the solution into two parts: *transient* and *steady state*.

We set $F_{\text{ext}} = 1\sin(t)$, and $c = 1$ for friction, which is fairly high for an oscillating system. Describe the resulting behaviour.

Specifically, what are the two natural frequencies in the solution?

Which frequency is present in the **long run**?

In a damped physical system with forced oscillations, is y_{NH} **transient** or **steady-state**, and why?

In a damped physical system, is y_c **transient** or **steady-state**, and why?

Reminder: we study the simple damped spring/mass system in such depth because

- the system displays all the interesting mathematical solution forms as we vary just a few parameters,
- we hope that the simplicity of the system, and its resemblance to familiar systems like swings or bouncing a basket ball, help to you associate the mathematics with the real-world behaviour.
- the DE for the spring/mass system is either identical or similar to those for a number of other real-world scenarios, like electronic circuits.

Unfortunately, that can all be a lot of work. Sometimes, we just want the prediction of y, position, velocity, etc. over time.

- In *principle*, the pair (DE + initial condition) is enough to define prediction for all time.
- The difficulties in finding analytic solutions come from the necessary integration/solution step.
- MATLAB can generate an *approximate numerical solution* **without** need to integrate!

Example - Pendulum

$$\begin{aligned} \text{Newton's Second Law: } mL^2\theta'' &= T_g + T_f \\ &= -mLg\sin(\theta) - (\mu L^2 m)\theta' \\ \text{Solving for } \theta''\text{: } \theta'' &= -\frac{g}{L}\sin(\theta) - \mu\theta' \end{aligned}$$

Problem. Turn this single second-order DE into a pair of first-order DEs.

Problem. Download the representation of this system of DEs, `pendulumDE.m`. Compare it to our DE system,

$$\frac{dy_1}{dt} = y_2$$
$$\frac{dy_2}{dt} = -\frac{g}{L}\sin(y_1) - \mu y_2$$

Problem. Download and run the script `W8_2.m`.

Problem. What were the initial conditions of the pendulum?

Problem. Based on that information, what do the two curves on the graph represent?

Problem. Change the script so the pendulum also has an initial velocity.

Problem. If we keep the initial angle at $-\frac{\pi}{2}$ (pendulum out horizontally), experiment and find the initial velocity that will push the pendulum "over the top".

Classifying Higher-Order DEs

While working with DEs in MATLAB, we haven't had to distinguish between types of equations: so long as they can be re-written in the form $\dfrac{dy}{dt} = f(t, y)$, MATLAB can produce a numerical solution or prediction.

However, when looking at references or mathematical derivations, where DEs are solved exactly using by-hand methods, the method used is usually chosen based on the form of the DE, so being able to categorize equations is a relevant skill.

Problem. Distinguish between the following classifications of differential equations, and given an example of each.

Linear vs Non-Linear

Linear Homogeneous and Linear Non-Homogeneous

Classify the following DEs based on the terms *homogeneous* and *linear*

$$x^2y'' + xy' + y = 10$$

$$100y'' + y = 4x^3$$

$$(y'')^2 + y = 4e^x$$

$$4y'' - 10y' + y = 0$$