HW12 Problems due Friday November 16 (**LAST DAY** before midterm)

12.1 CONCEPT: Characterizing ODEs (3 pts)

For each of the following three ODEs …

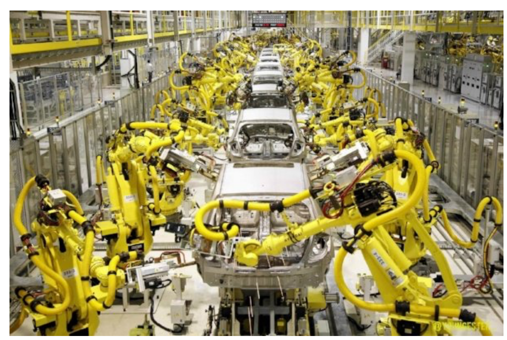
i. characterize its **order** (*e.g.* **1st**-order, **2nd**-order, etc.),
ii. characterize it as an **IVP** (Initial Value Problem) or **BVP** (Boundary Value Problem),
iii. write the ODE in **standard** form.

$$(A) \quad x + \frac{dy}{dx}\left(\frac{d^3y}{dx^3} + 1\right) = \frac{d^2y}{dx^2}, \quad y(1) = 1,\ y'(2) = 2,\ y''(1) = 3$$

$$(B) \quad t + y\sqrt{\frac{dt}{dy}} = t\log\left(\frac{d^2t}{dy^2}\right), \quad t(2) = 1,\ t'(1) = 2$$

$$(C) \quad 1 + \frac{1}{y}\frac{d\left(y^2 t\right)}{dt} = t\left(\frac{dy}{dt}\right), \quad y(5) = 2$$

Modeling Factory Production (IVP)



Let's evaluate the productivity of a factory by predicting the number of machines operating (**Y**) over the course of two months (60 days). On the first day ($\mathbf{t_0 = 0}$) you observe $\mathbf{Y_0 = 20}$ machines are operating.

Then Y will change over time (**dY / dt**, for **t** in days) according to the following 3 factors:

- The owner hopes to buy 3 new machines each day (dY/dt = +3),
- The machines have a natural failure rate (dY/dt = – 0.1 Y),
- External fluctuating factors affecting machine purchasing and operation (dY/dt = sin[2πt/25]).

Combining all these factors leads to the first-order IVP $\dfrac{dY}{dt} = 3 - 0.1Y + \sin\left(\dfrac{2\pi t}{45}\right)$ with I.C. $Y(0) = 20$

The goal of your analysis to make a plot of number of machines working each day **Y(t)**, and to specifically predict the number of machines working at the end of 2 months ($\mathbf{t_{end} = 60}$ days).

Let's tackle this two ways on the next page:

- *By-hand* comparing the Euler, Modified Euler (Heun's), and Midpoint method,
- Writing your own *MATLAB* code for <u>just</u> the Modified Euler method.

HW12 Problems due Friday November 16 (**LAST DAY** before midterm)

12.2 APPLICATION: Solving 1st-order IVP, "By-Hand" (14 pts)

$$\frac{dY}{dt} = 3 - 0.1Y + \sin\left(\frac{2\pi t}{45}\right)$$

$$Y(0) = 20$$

a) Solve the production model at right using the **EULER** method over the range t = [0, 20] days with step size Δt = 10 days (*i.e.* calculate Y_i at $t_0 = 0$, $t_1 = 10$, and $t_2 = 20$).

Show all your work in table form, like we did in class and seen below, showing how you calculate each t_{i+1}, Y_{i+1}, and "*slope*" that you use to get each point, with all values accurate to **three** decimals.

i	t_i	Y_i (known)	$Slope_i$ = …	t_{i+1}	Y_{i+1} = …
0	0	$Y_0 = 20$	?	?	Y_1 = …
1	10	Y_1 = …	?	?	Y_2 = …
2	20	Y_2 = …	DONE	DONE	DONE

b) Repeat the work from (a), this time using the **MODIFIED EULER** (Heun's) method to calculate Y_1, Y_2.

c) Repeat the work from (a), this time using **MIDPOINT** method to calculate Y_1, Y_2.

d) Given the exact solution for $Y(10) = 30.5795$, and $Y(20) = 34.8952$, calculate:

   i. the *global* error (to three decimals) in Y_2 for *just* the Euler method from part (a).

   ii. the *local* error for *just* the single step going from t_1 to t_2 for *just* the Euler method from part (a).

12.3 CODING: Solving the productivity model with your *own* Modified Euler method IN MATLAB (7 pts)

Go look in the Carmen HW12 folder. You have been given four files: `HW12.m`, `ModEuler.m` , `f.m` and `Y_exact.m`, each described below:

`f.m` outputs the right-hand side of the given factory production IVP in standard form, $dY/dt = f(t,Y)$:

```
function dYdt= f(t,Y)
  dYdt = 3 - 0.1*Y + sin(2*pi*t/45);
```

Do ***not*** change anything in this function. Notice that this function …

   i. is already "vectorized" (can accept vector inputs *t* and *Y*, and returns a vector output *dYdt*),

   ii. is a general function of *t* and *Y*,

   iii. the order of the input variables has the *independent* variable (*t*) first, and *dependent* variable (*Y*) second. This is <u>really</u> crucial later, when we're using MATLAB's built-in `ODE45` solver.

`ModEuler.m` is the function ***<u>you</u>*** must complete. It currently only has the function declaration line:

```
function [t,Y] = ModEuler(t0,tend,Y0,dt)
% Finish the code here to calculate the two output vectors
% t=[t0 t1 ... tend], Y=[Y0 Y1 ... ] given t0, tend, Y0 and dt.
```

`Y_exact.m` is also given to provide the exact, analytic solution to $Y(t)$ to compare to the ModEuler method later. It's really long – don't worry about how I got the analytic solution from your "real math" differential equations course. Just assume it's correct. ;)

HW12 Problems due Friday November 16 (**LAST DAY** before midterm)

12.3 continued …

`HW12.m` is the master script. Its job is to solve for the number of operational machines *Y*(*t*) with initial number Y_0 = 20 over the time range *t* = [0, 60] days using three different time steps: Δ*t* = 30 days, 3 days, and 0.3 days. It does this by calling your new function `ModEuler.m` three times like so:

```
[t1,Y1]= ModEuler(t0, tend, Y0, 30);     % Δt = 30 d
[t2,Y2]= ModEuler(t0, tend, Y0, 3);      % Δt = 3 d
[t3,Y3]= ModEuler(t0, tend, Y0, 0.3);    % Δt = 0.3 d
```

It also then calculates the exact (analytic) solutions and the error (difference between exact and computational solutions) over the three different time sets:

```
Y1x = Y_exact(t1);      Err1 = abs(Y1x-Y1);
Y2x = Y_exact(t2);      Err2 = abs(Y2x-Y2);
Y3x = Y_exact(t3);      Err3 = abs(Y3x-Y3);
```

Finally, HW12.m makes and saves two plots so you can compare the results for different Δ*t*:

- Plot **HW12a** compares the three different computational (t_i, Y_i) with the exact *Y*(*t*).
- Plot **HW12b** compares the computational errors in Y_i for the three choices of Δ*t*.

Much of the work has already been done for you. But I still need you to …

a) Complete the `ModEuler.m` function so that it uses the Modified Euler method to solve the IVP d*Y*/d*t* = f(*t*,*Y*), where f(*t*,*Y*) comes from the "slope" function `f.m` I gave you.

   Make sure `ModEuler.m` works with general input variables for `t0` (initial *t*), `tend` (final *t*), `Y0` (initial size), and `dt` (time step Δ*t*), like its declaration line shows. The function must output vectors for both the time (`t`) and the operating machines (`Y`) at each time-step from `t0` to `tend` inclusive.

b) Fill in the couple of missing variable names appropriately (currently `xxx`) in `HW12.m` so that it works again. Also replace the plot **`title`**s with more appropriate text for your plot. Other than that, don't mess much with `HW12.m` – it's already designed to work and make the plot for you just fine. If it gives you any problems, it's because you're making compatibility errors in the `ModEuler.m` function you're writing.

c) Run the HW12.m script to generate the two plots. They *should* show your Euler-method solutions more closely approaching the exact solution (in red) as Δ*t* gets smaller.

SUBMIT EVERYTHING FROM HW12.3 ONLINE:

- Both your plots (**HW12a.pdf**, **HW12b.pdf**) in .pdf form,
- Your functions **`ModEuler.m`** and **`HW12.m`**. (But don't submit your f.m or Y_exact.m, since you're not modifying them from what I already gave you).
- Your answers to the **three questions** below in the comment box:
  i. Report the predicted numer of operating machines **Y** at t = 60 days for each of the three simulations.
  ii. Report the ***global error*** at *t* = 21 days for just the last two simulations (using Δ*t* = 3 and 0.3 days).
  iii. How does the change between the two errors in (ii) compare with our <u>analytical expectation</u> based on the error "order" for the Modified Euler method? *Hint*: your Δ*t* went down by a factor of 10, from 3 to 0.3 days – by what factor did your observed error go down??)