Quick Questions

Use MATLAB to evaluate the following formulas. Show the results of executing the formulas in a MATLAB session. For each formula use variables, not numerical values for the constants given at the start of the problem statement. Pure numbers in the formulas can be entered directly.

Example:
Given \( m = 3 \) gm, \( v = 30 \) m/s, compute the kinetic energy \( KE = \frac{1}{2}mv^2 \)

Solution

\[
\begin{align*}
&>> m = 3e-3; \\
&>> v = 30; \\
&>> KE = 0.5*m*v^2 \\
&KE = \\
&1.3500
\end{align*}
\]

1. (5 points)

Given \( \rho = 1.2 \text{ kg/m}^3 \), \( V_0 = 5 \text{ m/s} \), \( \epsilon = 0.4 \Phi_s = 0.83 \), and \( D_p = 0.2 \text{ mm} \), compute \( \Delta p/L \)

\[
\Delta p \frac{L}{L} = \frac{1.75\rho V_0^2}{\Phi_s D_p} \frac{1 - \epsilon}{\epsilon^3}
\]

Make sure the units of \( \Delta p/L \) are Pa/m.

2. (5 points) Given \( p_a = 6.5 \times 10^5 \text{ Pa} \), \( p_b = 1.42 \times 10^5 \text{ Pa} \), \( \gamma = 1.4 \), and \( \beta = 0.75 \) compute \( Y \) where

\[
Y = \left( \frac{p_b}{p_a} \right)^{1/\gamma} \left\{ \frac{\gamma(1 - \beta^4)(1 - (p_b/p_a)^{(\gamma - 1)/\gamma})}{(\gamma - 1)(1 - p_b/p_a) \left[ 1 - \beta^4(p_b/p_a)^{2/\gamma} \right]} \right\}^{1/2}
\]
Comprehensive Questions

3. **(15 points)** The Michaelis-Mentel model for the consumption of a substrate by an enzymatic reaction is

\[ K_m \ln \left( \frac{s_0}{s} \right) + s_0 - s = V_{\text{max}}t \]

where \(K_m\) is the concentration of the substrate at the so-called *half-maximal consumption rate*, \(s\) is the substrate concentration, \(s_0\) is the initial value of \(s\), \(V_{\text{max}}\) is the maximum substrate consumption rate, and \(t\) is time.

Make a plot of \(s\) versus \(t\) for \(0 \leq t \leq 200\) minutes for the following parameter values: \(K_m = 0.5\) mM (milli-mole), \(s_0 = 1\) mM and \(V_{\text{max}} = 5\) mM/min. *Hint:* Since you cannot solve for \(s\) as a function of \(t\), compute \(t\) as a function of \(s\) and reverse the meaning of the abscissa and ordinate. \(t\) should still be the abscissa in your plot.

4. **(15 points)** Given the following astronomical data

- Diameter of the earth: \(d_e = 12,600\) km
- Diameter of the sun: \(d_s = 1,390,000\) km
- Orbit of the earth: \(d_o = 150,000,000\) km

use the `draw_circle` and `fill_circle` functions to make two plots. Assume that the sun, the earth, and the earth’s orbits are circular. The first plot shows (1) the sun as a solid yellow circle centered at the origin, (2) the orbit of the earth as a dashed line, and (3) the earth as a solid blue circle located on its orbit (choose one point).

The second plot is a zoomed-in view that shows the earth at the center of the plot and the roughly the size of one fifth of the \(x\) and \(y\) axes.

Be sure that the scale of the \(x\) and \(y\) axes are equal so that the sun, earth and earth orbits are circular.

*Do not* alter the code in `draw_circle` and `fill_circle` and *do not* copy the code from `draw_circle` and `fill_circle` into an m-file. Instead, write one m-file that calls `draw_circle` and `fill_circle` as needed.

Is there anything odd about the appearance of the dashed line representing the earth’s orbit. How would you fix the plot of the orbit to make it appear smooth?
function draw_circle(r,x0,y0,line_style)
% draw_circle Draw a circle in the (x,y) plane
% % Synopsis: draw_circle(r)
% % draw_circle(r,x0)
% % draw_circle(r,x0,y0)
% % Input: r = radius of the circle
% x0,y0 = x and y coordinates of the center of the circle
% Default: x0 = 0, y0 = 0;
if nargin<2, x0 = 0; end
if nargin<3, y0 = 0; end
if nargin<4, line_style = '-'; end

% t = linspace(0,2*pi);
x = x0 + r*cos(t);
y = y0 + r*sin(t);
plot(x,y,line_style)

function fill_circle(r,x0,y0,fill_style)
% fill_circle Draw a solid circle (a solid disk) in the (x,y) plane
% % Synopsis: fill_circle(r)
% % fill_circle(r,x0)
% % fill_circle(r,x0,y0)
% % Input: r = radius of the circle
% x0,y0 = x and y coordinates of the center of the circle
% Default: x0 = 0, y0 = 0;
if nargin<2, x0 = 0; end
if nargin<3, y0 = 0; end
if nargin<4, fill_style = 'b'; end

% t = linspace(0,2*pi);
x = x0 + r*cos(t);
y = y0 + r*sin(t);
fill(x,y,fill_style)