2.7 CASE STUDY continued

increases. Based on this result, you might conclude that your model needs to be improved. At the least, it might motivate you to conduct further experiments with a larger number of jumpers to confirm your preliminary finding.

In addition, the result might also stimulate you to go to the fluid mechanics literature and learn more about the science of drag. If you did this, you would discover that the parameter c_d is actually a lumped drag coefficient that along with the true drag includes other factors such as the jumper's frontal area and air density:

$$c_d = \frac{C_D \rho A}{2} \tag{2.2}$$

where C_D = a dimensionless drag coefficient, ρ = air density (kg/m^3), and A = frontal area (m^2), which is the area projected on a plane normal to the direction of the velocity.

Assuming that the densities were relatively constant during data collection (a pretty good assumption if the jumpers all took off from the same height on the same day), Eq. (2.2) suggests that heavier jumpers might have larger areas. This hypothesis could be substantiated by measuring the frontal areas of individuals of varying masses.

PROBLEMS

2.1 A simple electric circuit consisting of a resistor, a capacitor, and an inductor is depicted in Fig. P2.1. The charge on the capacitor $q(t)$ as a function of time can be computed as

$$q(t) = q_0 e^{-Rt/(2L)} \cos\left[\sqrt{\frac{1}{LC} - \left(\frac{R}{2L}\right)^2}\, t\right]$$

where t = time, q_0 = the initial charge, R = the resistance, L = inductance, and C = capacitance. Use MATLAB to generate a plot of this function from $t = 0$ to 0.7, given that $q_0 = 12$, $R = 50$, $L = 5$, and $C = 10^{-4}$.

2.2 The standard normal probability density function is a bell-shaped curve that can be represented as

$$f(z) = \frac{1}{\sqrt{2\pi}} e^{-z^2/2}$$

Use MATLAB to generate a plot of this function from $z = -4$ to 4. Label the ordinate as frequency and the abscissa as z.

2.3 Use the `linspace` function to create vectors identical to the following created with colon notation:

(a) `t = 5:6:30`

(b) `x = -3:4`

2.4 Use colon notation to create vectors identical to the following created with the `linspace` function:

(a) `v = linspace(-3,1,9)`

(b) `r = linspace(8,0,17)`

2.5 If a force F (N) is applied to compress a spring, its displacement x (m) can often be modeled by Hooke's law:

$$F = kx$$

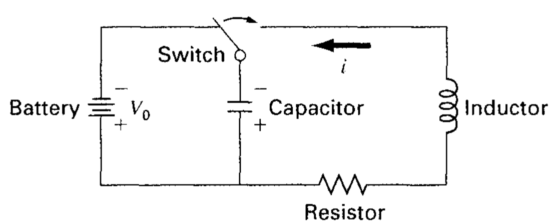

FIGURE P2.1

$$\frac{8}{15\sqrt{10} - 3\sqrt{14}} = \frac{4\sqrt{2}}{15\sqrt{5}\sqrt{2} - 3\sqrt{7}\sqrt{2}} = \frac{\sqrt{2}}{\sqrt{2}} \times \frac{4\sqrt{2}}{15\sqrt{5} - 3\sqrt{7}}$$

where k = the spring constant (N/m). The potential energy stored in the spring U (J) can then be computed as

$$U = \frac{1}{2}kx^2$$

Five springs are tested and the following data compiled:

***F*, N**	11	12	15	9	12
***x*, m**	0.013	0.020	0.009	0.010	0.012

Use MATLAB to store F and x as vectors and then compute vectors of the spring constants and the potential energies. Use the `max` function to determine the maximum potential energy.

2.6 The density of freshwater can be computed as a function of temperature with the following cubic equation:

$$\rho = 5.5289 \times 10^{-8}T_C^3 - 8.5016 \times 10^{-6}T_C^2 + 6.5622 \times 10^{-5}T_C + 0.99987$$

where ρ = density (g/cm^3) and T_C = temperature (°C). Use MATLAB to generate a vector of temperatures ranging from 32 °F to 82.4 °F using increments of 3.6 °F. Convert this vector to degrees Celsius and then compute a vector of densities based on the cubic formula. Create a plot of ρ versus T_C. Recall that $T_C = 5/9(T_F - 32)$.

2.7 Manning's equation can be used to compute the velocity of water in a rectangular open channel:

$$U = \frac{\sqrt{S}}{n}\left(\frac{BH}{B+2H}\right)^{2/3}$$

where U = velocity (m/s), S = channel slope, n = roughness coefficient, B = width (m), and H = depth (m). The following data is available for five channels:

n	*S*	*B*	*H*
0.035	0.0001	10	2
0.020	0.0002	8	1
0.015	0.0010	19	1.5
0.030	0.0008	24	3
0.022	0.0003	15	2.5

Store these values in a matrix where each row represents one of the channels and each column represents one of the parameters. Write a single-line MATLAB statement to compute a column vector containing the velocities based on the values in the parameter matrix.

2.8 It is general practice in engineering and science that equations be plotted as lines and discrete data as symbols. Here is some data for concentration (c) versus time (t) for the photodegradation of aqueous bromine:

***t*, min**	10	20	30	40	50	60
***c*, ppm**	3.4	2.6	1.6	1.3	1.0	0.5

This data can be described by the following function:

$$c = 4.84e^{-0.034t}$$

Use MATLAB to create a plot displaying both the data (using square symbols) and the function (using a dotted line). Plot the function for t = 0 to 75 min.

2.9 The `semilogy` function operates in an identical fashion to the `plot` function except that a logarithmic (base-10) scale is used for the y axis. Use this function to plot the data and function as described in Prob. 2.8. Explain the results.

2.10 Here is some wind tunnel data for force (F) versus velocity (v):

***v*, m/s**	10	20	30	40	50	60	70	80
***F*, N**	25	70	380	550	610	1220	830	1450

This data can be described by the following function:

$$F = 0.2741v^{1.9842}$$

Use MATLAB to create a plot displaying both the data (using diamond symbols) and the function (using a dotted line). Plot the function for v = 0 to 90 m/s.

2.11 The `loglog` function operates in an identical fashion to the `plot` function except that logarithmic scales are used for both the x and y axes. Use this function to plot the data and function as described in Prob. 2.10. Explain the results.

2.12 The Maclaurin series expansion for the sine is

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \cdots$$

Use MATLAB to create a plot of the sine (solid line) along with a plot of the series expansion (dashed line) up to and including the term $x^7/7!$. Use the built-in function `factorial` in computing the series expansion. Make the range of the abscissa from $x = 0$ to $3\pi/2$.

2.13 You contact the jumpers used to generate the data in Table 2.1 and measure their frontal areas. The resulting values, which are ordered in the same sequence as the corresponding values in Table 2.1, are

A, m²	0.454	0.401	0.453	0.485	0.532	0.474	0.486

(a) If the air density is $\rho = 1.225$ kg/m³, use MATLAB to compute values of the dimensionless drag coefficient C_D.
(b) Determine the average, minimum and maximum of the resulting values.
(c) Develop a side-by-side plot of A versus m (left side) and C_D versus m (right side). Include descriptive axis labels and titles on the plots.

2.14 The following parametric equations generate a helix that contracts exponentially as it evolves

$$x = e^{-0.1t}\sin t$$
$$y = e^{-0.1t}\cos t$$
$$z = t$$

Use `subplot` to generate a two-dimensional line plot of (x, y) in the top pane and a three-dimensional line plot of (x, y, z) in the bottom pane.

2.15 Exactly what will be displayed after the following MATLAB commands are typed?

(a)
```
>> x = 2;
>> x ^ 3;
>> y = 8 - x
```

(b)
```
>> q = 4:2:10;
>> r = [7 8 4; 3 6 -2];
>> sum(q) * r(2, 3)
```

2.16 The trajectory of an object can be modeled as

$$y = (\tan\theta_0)x - \frac{g}{2v_0^2\cos^2\theta_0}x^2 + y_0$$

where y = height (m), θ_0 = initial angle (radians), x = horizontal distance (m), g = gravitational acceleration (= 9.81 m/s²), v_0 = initial velocity (m/s), and y_0 = initial height. Use MATLAB to find the displacement for $y_0 = 0$ and $v_0 = 30$ m/s for initial angles ranging from 15 to 75° in increments of 15°. Employ a range of horizontal distances from $x = 0$ to 100 m in increments of 5 m. The results should be assembled in an array where the first dimension (rows) corresponds to the distances, and the second dimension (columns) corresponds to the different initial angles. Use this matrix to generate a single plot of the heights versus horizontal distances for each of the initial angles. Employ a legend to distinguish among the different cases, and scale the plot so that the minimum height is zero using the `axis` command.

2.17 The temperature dependence of chemical reactions can be computed with the *Arrhenius equation:*

$$k = Ae^{-E/(RT_a)}$$

where k = reaction rate (s⁻¹), A = the preexponential (or frequency) factor, E = activation energy (J/mol), R = gas constant [8.314 J/(mole · K)], and T_a = absolute temperature (K). A compound has $E = 1 \times 10^5$ J/mol and $A = 7 \times 10^{16}$. Use MATLAB to generate values of reaction rates for temperatures ranging from 273 to 333 K. Use `subplot` to generate a side-by-side graph of (a) k versus T_a and (b) $\log_{10} k$ versus $1/T_a$. Employ the `semilogy` function to create (b). Interpret your results.