Chapter 3: Engineering and Scientific Manipulations

3.1: Assignment (:=) and Expressions

Try It! (p. 48)

Express, for a general radius, the volume and surface area of a cone whose height is twice the radius.

Solution

```
> restart;
```

The volume and surface area of a cone with height *h* and base radius *r* are given by

```
> V := 1/3 * Pi * r^2 * h;
```

$$V := \frac{1}{3}\pi r^2 h$$

```
> S := Pi * r^2 + Pi * r * sqrt(r^2+h^2);
```

$$S := \pi r^2 + \pi r \sqrt{r^2 + h^2}$$

```
>
```

Since the height is twice the radius, we make the assignment

```
> h := 2 * r;
```

$$h := 2r$$

```
>
```

The volume and surface area are now seen to be

```
> volume = V;
```

$$volume = \frac{2}{3}\pi r^3$$

```
> surface_area = S;
```

$$surface\_area = \pi r^2 + \pi r \sqrt{5}\sqrt{r^2}$$

Note

Note that Maple does not automatically simplify $\sqrt{r^2}$. This is because Maple does not know that *r* is a positive quantity (it could be negative or complex valued). As you read further in this module you will learn several different ways to deal with this type of situation. Here are two possibilities:

```
> simplify( S, symbolic );
```

$$\pi r^2 + \pi r^2 \sqrt{5}$$

```
>
> assume( r > 0 );
> simplify( S );
```

$$\pi r\sim^2 + \pi r\sim^2 \sqrt{5}$$

```
>
```

3.2: Expression Sequences, Lists and Sets

Try It! (p. 52)

Suppose you want to digitize an analog voice signal, which ranges from 0 mVolts to 50 mVolts in such a manner as to use binary bits (0s or 1s). You decide that quantizing the amplitude level into 128 discrete and equal-width intervals over the range of 0 to 50 mVolts will be sufficient. Use the seq command to generate a list of the 128 levels that will be represented by these binary codes.

Solution

```
> restart;
```

The basic idea is simply to divide the interval [0, 50] into 128 equal-sized subintervals; this requires 129 evenly spaced points from the interval [0,50]:

```
> QUANT := [ seq( 50*(i/128), i=0..128 ) ];
```

$$QUANT := \left[0, \frac{25}{64}, \frac{25}{32}, \frac{75}{64}, \frac{25}{16}, \frac{125}{64}, \frac{75}{32}, \frac{175}{64}, \frac{25}{8}, \frac{225}{64}, \frac{125}{32}, \frac{275}{64}, \frac{75}{16}, \frac{325}{64}, \frac{175}{32}, \frac{375}{64}, \frac{25}{4}, \frac{425}{64}, \frac{225}{32}, \frac{475}{64}, \frac{125}{16}, \frac{525}{64}, \frac{275}{32},\right.$$

$$\frac{575}{64}, \frac{75}{8}, \frac{625}{64}, \frac{325}{32}, \frac{675}{64}, \frac{175}{16}, \frac{725}{64}, \frac{375}{32}, \frac{775}{64}, \frac{25}{2}, \frac{825}{64}, \frac{425}{32}, \frac{875}{64}, \frac{225}{16}, \frac{925}{64}, \frac{475}{32}, \frac{975}{64}, \frac{125}{8}, \frac{1025}{64}, \frac{525}{32}, \frac{1075}{64}, \frac{275}{16}, \frac{1125}{64},$$
$$\frac{575}{32}, \frac{1175}{64}, \frac{75}{4}, \frac{1225}{64}, \frac{625}{32}, \frac{1275}{64}, \frac{325}{16}, \frac{1325}{64}, \frac{675}{32}, \frac{1375}{64}, \frac{175}{8}, \frac{1425}{64}, \frac{725}{32}, \frac{1475}{64}, \frac{375}{16}, \frac{1525}{64}, \frac{775}{32}, \frac{1575}{64}, 25, \frac{1625}{64}, \frac{825}{32},$$
$$\frac{1675}{64}, \frac{425}{16}, \frac{1725}{64}, \frac{875}{32}, \frac{1775}{64}, \frac{225}{8}, \frac{1825}{64}, \frac{925}{32}, \frac{1875}{64}, \frac{475}{16}, \frac{1925}{64}, \frac{975}{32}, \frac{1975}{64}, \frac{125}{4}, \frac{2025}{64}, \frac{1025}{32}, \frac{2075}{64}, \frac{525}{16}, \frac{2125}{64}, \frac{1075}{32},$$
$$\frac{2175}{64}, \frac{275}{8}, \frac{2225}{64}, \frac{1125}{32}, \frac{2275}{64}, \frac{575}{16}, \frac{2325}{64}, \frac{1175}{32}, \frac{2375}{64}, \frac{75}{2}, \frac{2425}{64}, \frac{1225}{32}, \frac{2475}{64}, \frac{625}{16}, \frac{2525}{64}, \frac{1275}{32}, \frac{2575}{64}, \frac{325}{8}, \frac{2625}{64},$$
$$\frac{1325}{32}, \frac{2675}{64}, \frac{675}{16}, \frac{2725}{64}, \frac{1375}{32}, \frac{2775}{64}, \frac{175}{4}, \frac{2825}{64}, \frac{1425}{32}, \frac{2875}{64}, \frac{725}{16}, \frac{2925}{64}, \frac{1475}{32}, \frac{2975}{64}, \frac{375}{8}, \frac{3025}{64}, \frac{1525}{32}, \frac{3075}{64}, \frac{775}{16},$$
$$\left.\frac{3125}{64}, \frac{1575}{32}, \frac{3175}{64}, 50\right]$$

Or, if floating-point numbers are preferred,

```
> QUANTf := [ seq( trunc( 50.*(i/128)*100 )/100., i=0..128 ) ];
```

QUANTf := [0, .3900000000, .7800000000, 1.170000000, 1.560000000, 1.950000000, 2.340000000, 2.730000000, 3.120000000, 3.510000000, 3.900000000, 4.290000000, 4.680000000, 5.070000000, 5.460000000, 5.850000000, 6.250000000, 6.640000000, 7.030000000, 7.420000000, 7.810000000, 8.200000000, 8.590000000, 8.980000000, 9.370000000, 9.760000000, 10.15000000, 10.54000000, 10.93000000, 11.32000000, 11.71000000, 12.10000000, 12.50000000, 12.89000000, 13.28000000, 13.67000000, 14.06000000, 14.45000000, 14.84000000, 15.23000000, 15.62000000, 16.01000000, 16.40000000, 16.79000000, 17.18000000, 17.57000000, 17.96000000, 18.35000000, 18.75000000, 19.14000000, 19.53000000, 19.92000000, 20.31000000, 20.70000000, 21.09000000, 21.48000000, 21.87000000, 22.26000000, 22.65000000, 23.04000000, 23.43000000, 23.82000000, 24.21000000, 24.60000000, 25.00000000, 25.39000000, 25.78000000, 26.17000000, 26.56000000, 26.95000000, 27.34000000, 27.73000000, 28.12000000, 28.51000000, 28.90000000, 29.29000000, 29.68000000, 30.07000000, 30.46000000, 30.85000000, 31.25000000, 31.64000000, 32.03000000, 32.42000000, 32.81000000, 33.20000000, 33.59000000, 33.98000000, 34.37000000, 34.76000000, 35.15000000, 35.54000000, 35.93000000, 36.32000000, 36.71000000, 37.10000000, 37.50000000, 37.89000000, 38.28000000, 38.67000000, 39.06000000, 39.45000000, 39.84000000, 40.23000000, 40.62000000, 41.01000000, 41.40000000, 41.79000000, 42.18000000, 42.57000000, 42.96000000, 43.35000000, 43.75000000, 44.14000000, 44.53000000, 44.92000000, 45.31000000, 45.70000000, 46.09000000, 46.48000000, 46.87000000, 47.26000000, 47.65000000, 48.04000000, 48.43000000, 48.82000000, 49.21000000, 49.60000000, 50.00000000]

```
>
```

An alternate method of obtaining the floating-point solution is introduced later in this chapter.

```
> QUANTf := evalf( QUANTf, 4 );
```

QUANTf := [0, .3900, .7800, 1.170, 1.560, 1.950, 2.340, 2.730, 3.120, 3.510, 3.900, 4.290, 4.680, 5.070, 5.460, 5.850, 6.250, 6.640, 7.030, 7.420, 7.810, 8.200, 8.590, 8.980, 9.370, 9.760, 10.15, 10.54, 10.93, 11.32, 11.71, 12.10, 12.50, 12.89, 13.28, 13.67, 14.06, 14.45, 14.84, 15.23, 15.62, 16.01, 16.40, 16.79, 17.18, 17.57, 17.96, 18.35, 18.75, 19.14, 19.53, 19.92, 20.31, 20.70, 21.09, 21.48, 21.87, 22.26, 22.65, 23.04, 23.43, 23.82, 24.21, 24.60, 25.00, 25.39, 25.78, 26.17, 26.56, 26.95, 27.34, 27.73, 28.12, 28.51, 28.90, 29.29, 29.68, 30.07, 30.46, 30.85, 31.25, 31.64, 32.03, 32.42, 32.81, 33.20, 33.59, 33.98, 34.37, 34.76, 35.15, 35.54, 35.93, 36.32, 36.71, 37.10, 37.50, 37.89, 38.28, 38.67, 39.06, 39.45, 39.84, 40.23, 40.62, 41.01, 41.40, 41.79, 42.18, 42.57, 42.96, 43.35, 43.75, 44.14, 44.53, 44.92, 45.31, 45.70, 46.09, 46.48, 46.87, 47.26, 47.65, 48.04, 48.43, 48.82, 49.21, 49.60, 50.00]

```
>
```

The intervals are formed from pairs of consecutive elements of the list:

```
> INTERVALS := [ seq( [QUANT[i],QUANT[i+1]], i=1..128 ) ];
```

$$INTERVALS := \left[\left[0, \frac{25}{64}\right], \left[\frac{25}{64}, \frac{25}{32}\right], \left[\frac{25}{32}, \frac{75}{64}\right], \left[\frac{75}{64}, \frac{25}{16}\right], \left[\frac{25}{16}, \frac{125}{64}\right], \left[\frac{125}{64}, \frac{75}{32}\right], \left[\frac{75}{32}, \frac{175}{64}\right], \left[\frac{175}{64}, \frac{25}{8}\right], \left[\frac{25}{8}, \frac{225}{64}\right],\right.$$
$$\left[\frac{225}{64}, \frac{125}{32}\right], \left[\frac{125}{32}, \frac{275}{64}\right], \left[\frac{275}{64}, \frac{75}{16}\right], \left[\frac{75}{16}, \frac{325}{64}\right], \left[\frac{325}{64}, \frac{175}{32}\right], \left[\frac{175}{32}, \frac{375}{64}\right], \left[\frac{375}{64}, \frac{25}{4}\right], \left[\frac{25}{4}, \frac{425}{64}\right], \left[\frac{425}{64}, \frac{225}{32}\right],$$
$$\left[\frac{225}{32}, \frac{475}{64}\right], \left[\frac{475}{64}, \frac{125}{16}\right], \left[\frac{125}{16}, \frac{525}{64}\right], \left[\frac{525}{64}, \frac{275}{32}\right], \left[\frac{275}{32}, \frac{575}{64}\right], \left[\frac{575}{64}, \frac{75}{8}\right], \left[\frac{75}{8}, \frac{625}{64}\right], \left[\frac{625}{64}, \frac{325}{32}\right], \left[\frac{325}{32}, \frac{675}{64}\right],$$
$$\left[\frac{675}{64}, \frac{175}{16}\right], \left[\frac{175}{16}, \frac{725}{64}\right], \left[\frac{725}{64}, \frac{375}{32}\right], \left[\frac{375}{32}, \frac{775}{64}\right], \left[\frac{775}{64}, \frac{25}{2}\right], \left[\frac{25}{2}, \frac{825}{64}\right], \left[\frac{825}{64}, \frac{425}{32}\right], \left[\frac{425}{32}, \frac{875}{64}\right], \left[\frac{875}{64}, \frac{225}{16}\right],$$
$$\left[\frac{225}{16}, \frac{925}{64}\right], \left[\frac{925}{64}, \frac{475}{32}\right], \left[\frac{475}{32}, \frac{975}{64}\right], \left[\frac{975}{64}, \frac{125}{8}\right], \left[\frac{125}{8}, \frac{1025}{64}\right], \left[\frac{1025}{64}, \frac{525}{32}\right], \left[\frac{525}{32}, \frac{1075}{64}\right], \left[\frac{1075}{64}, \frac{275}{16}\right],$$
$$\left[\frac{275}{16}, \frac{1125}{64}\right], \left[\frac{1125}{64}, \frac{575}{32}\right], \left[\frac{575}{32}, \frac{1175}{64}\right], \left[\frac{1175}{64}, \frac{75}{4}\right], \left[\frac{75}{4}, \frac{1225}{64}\right], \left[\frac{1225}{64}, \frac{625}{32}\right], \left[\frac{625}{32}, \frac{1275}{64}\right], \left[\frac{1275}{64}, \frac{325}{16}\right],$$
$$\left[\frac{325}{16}, \frac{1325}{64}\right], \left[\frac{1325}{64}, \frac{675}{32}\right], \left[\frac{675}{32}, \frac{1375}{64}\right], \left[\frac{1375}{64}, \frac{175}{8}\right], \left[\frac{175}{8}, \frac{1425}{64}\right], \left[\frac{1425}{64}, \frac{725}{32}\right], \left[\frac{725}{32}, \frac{1475}{64}\right], \left[\frac{1475}{64}, \frac{375}{16}\right],$$
$$\left[\frac{375}{16}, \frac{1525}{64}\right], \left[\frac{1525}{64}, \frac{775}{32}\right], \left[\frac{775}{32}, \frac{1575}{64}\right], \left[\frac{1575}{64}, 25\right], \left[25, \frac{1625}{64}\right], \left[\frac{1625}{64}, \frac{825}{32}\right], \left[\frac{825}{32}, \frac{1675}{64}\right], \left[\frac{1675}{64}, \frac{425}{16}\right],$$
$$\left[\frac{425}{16}, \frac{1725}{64}\right], \left[\frac{1725}{64}, \frac{875}{32}\right], \left[\frac{875}{32}, \frac{1775}{64}\right], \left[\frac{1775}{64}, \frac{225}{8}\right], \left[\frac{225}{8}, \frac{1825}{64}\right], \left[\frac{1825}{64}, \frac{925}{32}\right], \left[\frac{925}{32}, \frac{1875}{64}\right], \left[\frac{1875}{64}, \frac{475}{16}\right],$$
$$\left[\frac{475}{16}, \frac{1925}{64}\right], \left[\frac{1925}{64}, \frac{975}{32}\right], \left[\frac{975}{32}, \frac{1975}{64}\right], \left[\frac{1975}{64}, \frac{125}{4}\right], \left[\frac{125}{4}, \frac{2025}{64}\right], \left[\frac{2025}{64}, \frac{1025}{32}\right], \left[\frac{1025}{32}, \frac{2075}{64}\right], \left[\frac{2075}{64}, \frac{525}{16}\right],$$
$$\left[\frac{525}{16}, \frac{2125}{64}\right], \left[\frac{2125}{64}, \frac{1075}{32}\right], \left[\frac{1075}{32}, \frac{2175}{64}\right], \left[\frac{2175}{64}, \frac{275}{8}\right], \left[\frac{275}{8}, \frac{2225}{64}\right], \left[\frac{2225}{64}, \frac{1125}{32}\right], \left[\frac{1125}{32}, \frac{2275}{64}\right], \left[\frac{2275}{64}, \frac{575}{16}\right],$$
$$\left[\frac{575}{16}, \frac{2325}{64}\right], \left[\frac{2325}{64}, \frac{1175}{32}\right], \left[\frac{1175}{32}, \frac{2375}{64}\right], \left[\frac{2375}{64}, \frac{75}{2}\right], \left[\frac{75}{2}, \frac{2425}{64}\right], \left[\frac{2425}{64}, \frac{1225}{32}\right], \left[\frac{1225}{32}, \frac{2475}{64}\right], \left[\frac{2475}{64}, \frac{625}{16}\right],$$
$$\left[\frac{625}{16}, \frac{2525}{64}\right], \left[\frac{2525}{64}, \frac{1275}{32}\right], \left[\frac{1275}{32}, \frac{2575}{64}\right], \left[\frac{2575}{64}, \frac{325}{8}\right], \left[\frac{325}{8}, \frac{2625}{64}\right], \left[\frac{2625}{64}, \frac{1325}{32}\right], \left[\frac{1325}{32}, \frac{2675}{64}\right], \left[\frac{2675}{64}, \frac{675}{16}\right],$$
$$\left[\frac{675}{16}, \frac{2725}{64}\right], \left[\frac{2725}{64}, \frac{1375}{32}\right], \left[\frac{1375}{32}, \frac{2775}{64}\right], \left[\frac{2775}{64}, \frac{175}{4}\right], \left[\frac{175}{4}, \frac{2825}{64}\right], \left[\frac{2825}{64}, \frac{1425}{32}\right], \left[\frac{1425}{32}, \frac{2875}{64}\right], \left[\frac{2875}{64}, \frac{725}{16}\right],$$
$$\left[\frac{725}{16}, \frac{2925}{64}\right], \left[\frac{2925}{64}, \frac{1475}{32}\right], \left[\frac{1475}{32}, \frac{2975}{64}\right], \left[\frac{2975}{64}, \frac{375}{8}\right], \left[\frac{375}{8}, \frac{3025}{64}\right], \left[\frac{3025}{64}, \frac{1525}{32}\right], \left[\frac{1525}{32}, \frac{3075}{64}\right], \left[\frac{3075}{64}, \frac{775}{16}\right],$$
$$\left.\left[\frac{775}{16}, \frac{3125}{64}\right], \left[\frac{3125}{64}, \frac{1575}{32}\right], \left[\frac{1575}{32}, \frac{3175}{64}\right], \left[\frac{3175}{64}, 50\right]\right]$$

```
>
Or, if you prefer the floating-point version,
> INTERVALSf := [ seq( [QUANTf[i],QUANTf[i+1]], i=1..128 ) ];
```

INTERVALSf := [[0, .3900], [.3900, .7800], [.7800, 1.170], [1.170, 1.560], [1.560, 1.950], [1.950, 2.340],
[2.340, 2.730], [2.730, 3.120], [3.120, 3.510], [3.510, 3.900], [3.900, 4.290], [4.290, 4.680], [4.680, 5.070],
[5.070, 5.460], [5.460, 5.850], [5.850, 6.250], [6.250, 6.640], [6.640, 7.030], [7.030, 7.420], [7.420, 7.810],
[7.810, 8.200], [8.200, 8.590], [8.590, 8.980], [8.980, 9.370], [9.370, 9.760], [9.760, 10.15], [10.15, 10.54],
[10.54, 10.93], [10.93, 11.32], [11.32, 11.71], [11.71, 12.10], [12.10, 12.50], [12.50, 12.89], [12.89, 13.28],
[13.28, 13.67], [13.67, 14.06], [14.06, 14.45], [14.45, 14.84], [14.84, 15.23], [15.23, 15.62], [15.62, 16.01],
[16.01, 16.40], [16.40, 16.79], [16.79, 17.18], [17.18, 17.57], [17.57, 17.96], [17.96, 18.35], [18.35, 18.75],
[18.75, 19.14], [19.14, 19.53], [19.53, 19.92], [19.92, 20.31], [20.31, 20.70], [20.70, 21.09], [21.09, 21.48],
[21.48, 21.87], [21.87, 22.26], [22.26, 22.65], [22.65, 23.04], [23.04, 23.43], [23.43, 23.82], [23.82, 24.21],
[24.21, 24.60], [24.60, 25.00], [25.00, 25.39], [25.39, 25.78], [25.78, 26.17], [26.17, 26.56], [26.56, 26.95],
[26.95, 27.34], [27.34, 27.73], [27.73, 28.12], [28.12, 28.51], [28.51, 28.90], [28.90, 29.29], [29.29, 29.68],

$[29.68, 30.07], [30.07, 30.46], [30.46, 30.85], [30.85, 31.25], [31.25, 31.64], [31.64, 32.03], [32.03, 32.42],$
$[32.42, 32.81], [32.81, 33.20], [33.20, 33.59], [33.59, 33.98], [33.98, 34.37], [34.37, 34.76], [34.76, 35.15],$
$[35.15, 35.54], [35.54, 35.93], [35.93, 36.32], [36.32, 36.71], [36.71, 37.10], [37.10, 37.50], [37.50, 37.89],$
$[37.89, 38.28], [38.28, 38.67], [38.67, 39.06], [39.06, 39.45], [39.45, 39.84], [39.84, 40.23], [40.23, 40.62],$
$[40.62, 41.01], [41.01, 41.40], [41.40, 41.79], [41.79, 42.18], [42.18, 42.57], [42.57, 42.96], [42.96, 43.35],$
$[43.35, 43.75], [43.75, 44.14], [44.14, 44.53], [44.53, 44.92], [44.92, 45.31], [45.31, 45.70], [45.70, 46.09],$
$[46.09, 46.48], [46.48, 46.87], [46.87, 47.26], [47.26, 47.65], [47.65, 48.04], [48.04, 48.43], [48.43, 48.82],$
$[48.82, 49.21], [49.21, 49.60], [49.60, 50.00]]$

```
>
```

3.3: Creation and Dissection of Equations

Try It! (p. 53)

The equations found in Example 3-9 are not defined for certain combinations of the points $(x0, y0)$ and $(x1, y1)$. Use Maple to manipulate LINE into an equivalent form that does not involve fractions.

Hint

(The numerator and denominator of a fraction can be accessed via the numer and denom functions. Use the on-line help to determine how to use numer and denom.)

Solution

```
> restart;
```

Example 3-9 introduced the following two equations for a line through two given points:

```
> LINE := (y-y0)/(x-x0) = (y1-y0)/(x1-x0);
```

$$LINE := \frac{y-y0}{x-x0} = \frac{y1-y0}{x1-x0}$$

```
> LINE1 := lhs(LINE)/rhs(LINE) = 1;
```

$$LINE1 := \frac{(y-y0)\,(x1-x0)}{(x-x0)\,(y1-y0)} = 1$$

```
>
```

An equivalent equation which will not suffer from possible division by zero errors is:

```
> LINE2 := numer(lhs(LINE1)) = denom(lhs(LINE1));
```

$$LINE3 := (-y+y0)\,(x1-x0) = (-x+x0)\,(y1-y0)$$

```
>
```

Note that Maple sometimes re-orders terms in an expression (e.g., $-y+y0$ in this solution).

3.4: Solving Equations and Systems of Equations

Try It! (p. 54)

The quadratic equation $a\,x^2+b\,x+c=0$ has two solutions. Use solve to find these solutions, and then verify that these solutions are consistent with the quadratic formula. When a>0, b>0, and c<0 the discriminant $b^2-4\,ac$ is positive and larger than b^2. Thus, there will be one positive root and one negative root. Assign the positive root to the name POS and the negative root to the name NEG.

Solution

```
> restart;
```

The general form of a quadratic equation is

```
> EQN := a*x^2 + b*x + c = 0;
```

$$EQN := a\,x^2+b\,x+c=0$$

```
>
```

The two roots of the quadratic are:

```
> ROOTS := solve( EQN, { x } );
```

$$ROOTS := \{x = \frac{1}{2}\,\frac{-b+\sqrt{b^2-4\,a\,c}}{a}\},\ \{x = \frac{1}{2}\,\frac{-b-\sqrt{b^2-4\,a\,c}}{a}\}$$

>

It is clear that these solutions are consistent with the standard quadratic formula. The positive root is

```
> POS := op( ROOTS[1] );
```

$$POS := x = \frac{1}{2}\,\frac{-b+\sqrt{b^2-4\,a\,c}}{a}$$

and the negative root is

```
> NEG := op( ROOTS[2] );
```

$$NEG := x = \frac{1}{2}\,\frac{-b-\sqrt{b^2-4\,a\,c}}{a}$$

Note

The results of solve may be displayed in any order. If the two solutions are displayed in the opposite order you will need to interchange the assignments to POS and NEG.

>

3.5: Substitution and Evaluation

Try It! (p. 57)

Determine when the two roots of the quadratic equation differ by a constant Δ. When are the roots equal ($\Delta = 0$)?

Solution

```
> restart;
```

We begin by repeating the definitions from the previous Try It! (p. 54).

```
> EQN := a*x^2 + b*x + c = 0;
```

$$EQN := a\,x^2 + b\,x + c = 0$$

```
> ROOTS := solve( EQN, { x } );
```

$$ROOTS := \{x = \frac{1}{2}\,\frac{-b+\sqrt{b^2-4\,a\,c}}{a}\},\ \{x = \frac{1}{2}\,\frac{-b-\sqrt{b^2-4\,a\,c}}{a}\}$$

The magnitude of the difference between the roots is

```
> DIFF := simplify( rhs(op(ROOTS[1]))-rhs(op(ROOTS[2])) );
```

$$DIFF := \frac{\sqrt{b^2-4\,a\,c}}{a}$$

and so the roots differ by Δ when the constants a, b, and c are chosen so that

```
> CONST := solve( DIFF = Delta, { a, b, c } );
```

$$CONST := \{b = b,\ a = a,\ c = -\frac{1}{4}\,\frac{-b^2+\Delta^2\,a^2}{a}\}$$

>

That is, for any values of the constants a and b, the third constant c should be chosen so that $c = \frac{b^2-\Delta^2\,a^2}{4\,a}$. In particular, with $\Delta = 0$ this is seen to reduce to $c = \frac{b^2}{4\,a}$.

There is no reason to use Maple for this simplification. But, if you insist

```
> subs( Delta=0, CONST );
```

$$\{b = b,\ c = \frac{1}{4}\,\frac{b^2}{a},\ a = a\}$$

Note: complex values

Note that DIFF is a complex number when $\sqrt{b^2-4\,a\,c}$ is negative. This is one reason why many engineering applications are more interested in the magnitude of DIFF. (Another reason to consider $|DIFF|$ is that the order of the roots returned by solve cannot be taken for granted.)

```
> 
> solve( abs(DIFF)=Delta, { a, b, c } );
```

$$\{c=-\frac{1}{4}\frac{-b^2+\Delta^2 a^2}{a}, a=a, b=b\}, \{c=-\frac{1}{4}\frac{-b^2+\Delta^2 a^2}{a}, a=a, b=b\}$$

Observe that while Maple appears to have found two solutions, they are the same -- and the same as was found previously.

Note: alternate syntax for solve

The definition of DIFF is unnecessarily complicated. If the second argument in the solve command in the definition of ROOTS is changed from { x }, to x, it would be possible to use

```
> ROOTS := solve( EQN, x );
```

$$ROOTS := \frac{1}{2}\frac{-b+\sqrt{b^2-4\,a\,c}}{a}, \frac{1}{2}\frac{-b-\sqrt{b^2-4\,a\,c}}{a}$$

```
> DIFF := simplify( ROOTS[1]-ROOTS[2] );
```

$$DIFF := \frac{\sqrt{b^2-4\,a\,c}}{a}$$

```
> 
> 
```

Try It! (p. 59)

Predict, and explain, the results produced by the following two commands.

```
> TEST1 := subs( { x = y, y = x }, [x,y] ):
> TEST2 := subs( x=y, y=x, [x,y] ):
```

Hint

Consult the help worksheet for subs.

Solutions

```
> restart;
```

Two methods of accessing the online help for subs can be found by following the hyperlink in this sentence or by executing the following command

```
> ?subs
> 
```

The first example simultaneously replaces x with y and y with x, i.e. x and y are interchanged.

```
> TEST1 := subs( { x = y, y = x }, [x,y] );
```

$$TEST1 := [y, x]$$

```
> 
```

The substitutions in TEST2 are applied in succession. Thus, the first step is equivalent to

```
> TEST2a := subs( x=y, [x,y] );
```

$$TEST2a := [y, y]$$

followed by

```
> TEST2b := subs( y=x, TEST2a );
```

$$TEST2b := [x, x]$$

```
> TEST2 := subs( x=y, y=x, [x,y] );
```

$$TEST2 := [x, x]$$

```
> 
```

What If? (p.66)

Budget constraints require smaller engines with less thrust. Thus, less drag can be accommodated. Find an explicit formula that expresses the thrust in terms of the weight, lift coefficient, and other parameters (excluding the drag coefficient). What happens to the thrust requirement as α decreases? What do you think can be done to the airplane design to reduce the constant α in the expression $C_D = C_{D_0} + \alpha\, C_L^{\,2}$?

Solution

```
> restart;
```

The explicit formula for the thrust can be derived from the two balance laws, the definitions of lift and drag, and the lift-to-drag equation:

```
> balance1 := lift = weight;
```

$$balance1 := \frac{1}{2}\rho V^2 S\, CL = weight$$

```
> balance2 := thrust = drag;
```

$$balance2 := thrust = \frac{1}{2}\rho V^2 S\, CD$$

```
> lift := rho*V^2/2 * S * CL;
```

$$lift := \frac{1}{2}\rho V^2 S\, CL$$

```
> drag := rho*V^2/2 * S * CD;
```

$$drag := \frac{1}{2}\rho V^2 S\, CD$$

```
> liftdrag := CD = CD0 + alpha*CL^2;
```

$$liftdrag := CD = CD0 + \alpha\, CL^2$$

```
>
```

The first step is to substitute the lift-to-drag equation into the balance equation for thrust:

```
> subs( liftdrag, balance2 );
```

$$thrust = \frac{1}{2}\rho V^2 S\,(CD0 + \alpha\, CL^2)$$

Note that this is an explicit formula for the thrust that does not depend on C_D. However, it also does not depend on the weight. To introduce the weight as a variable in the formula for thrust, balance1 must be used. To ensure that the substitution is successful, it is recommended the solve this equation for one of the variables in the leading coefficient for the thrust:

```
> subs( solve(balance1,{S}), " );
```

$$thrust = \frac{weight\,(CD0 + \alpha\, CL^2)}{CL}$$

Or, in a slightly different form:

```
> collect(", {CL,weight});
```

$$thrust = \left(\alpha\, CL + \frac{CD0}{CL}\right) weight$$

```
>
```

From this formula for the thrust it is easily seen that the thrust decreases as α decreases. (In fact, the thrust is a linear function in the parameter α.)

To decrease the value of α the airplane will need to have a lower C_D for any given C_L. This can be achieved by making the plane more aerodynamic.

```
>
```

3.6: Functions

Try It! (p. 67)

The first step towards defining functions in Maple is to realize that the command `g(x) := x^2;` defines only the name `g(x)`. Verify that the name `g(x);` returns the expression x^2, but `g(0);`, `g(y);`, and `g(2*x);` all return unevaluated.

Solution

```
> restart;
```

Here is the given definition.

```
> g(x) := x^2;
```

$$g(x) := x^2$$

```
>
```

The results of the four commands are as follows:

```
> g(x);
```

$$x^2$$

```
> g(0);
```

$$g(0)$$

```
> g(y);
```

$$g(y)$$

```
> g(2*x);
```

$$g(2\,x)$$

As expected, all results are returned unevaluated except `g(x)`.

```
>
```

Try It! (p. 69)

Use the data from the Application 3 to create a function that can be used to obtain the drag for any value of the coefficient of lift.

Solution

```
> restart;
```

The drag, the relationship between the coefficients of lift and drag, and the other data (given and computed) needed to solve this problem are

```
> drag := gamma*delta*Psl*M^2*b^2*C[D]/(2*AR);
```

$$drag := \frac{1}{2}\frac{\gamma\,\delta\,Psl\,M^2\,b^2\,C_D}{AR}$$

```
> drag := rho*V^2*S*CD/2;
```

$$drag := \frac{1}{2}\rho\,V^2\,S\,CD$$

```
> liftdrag := CD = CD0+alpha*CL^2;
```

$$liftdrag := CD = CD0 + \alpha\,CL^2$$

```
> PARAM := evalf( [w = 500000, b = 200, AR = 10, M = 0.84, gamma1 = 1.4, p0 =
  14.696*12^2, delta = .2360, rho0 = 0.002377, sigma = 0.3106 ], 4 );
```

$$PARAM := [w = 500000., b = 200., AR = 10., M = .84, \gamma 1 = 1.4, p0 = 2116., \delta = .2360, \rho 0 = .002377, \sigma = .3106]$$

```
> VARS := [ weight=w, V=M*a, S=b^2/AR, rho=sigma*rho0 ];
```

$$VARS := \left[weight = w, V = M\,a, S = \frac{b^2}{AR}, \rho = \sigma\,\rho 0 \right]$$

```
> Vsound := subs( [ p=delta*p0, rho=sigma*rho0], a=sqrt(p/rho*gamma1) );
```

$$Vsound := a = \sqrt{\frac{\delta\,p0\,\gamma 1}{\sigma\,\rho 0}}$$

```
> coefL2 := CL = 0.5066;
```

$$coefL2 := CL = .5066$$

```
> LDcoef := { CD0 = 0.01691523810, alpha = 0.04990476190 };
```

$$LDcoef := \{ CD0 = .01691523810, \alpha = .04990476190 \}$$

```
>
```

To express the drag as a function of C_L requires the substitution of the lift-to-drag equation and the other parameters into the expression for drag prior to creating a function from the resulting expression. Thus,

```
> DRAG := unapply( subs( liftdrag, VARS, Vsound, PARAM, LDcoef, drag ), CL );
```

$$DRAG := CL \to 16688.69528 + 49236.39618\,CL^2$$

```
>
```

As an application of the use of this function, observe that the thrust corresponding to level flight is, therefore,

```
> DRAG(rhs(coefL2));
```

$$29324.89928$$

This result is consistent with the result obtained in Step 4 of the application (p. 66).

```
>
```

3.7: Exact vs. Approximate Arithmetic

Try It! (p. 70)

Use subs to substitute the values in exact, default, three, and round3 into $x^2 - 3x - 1$. How many digits of accuracy are obtained with each set of solutions?

Solution

```
> restart;
```

The definition of R, and the different representations of the solution to the given polynomial, are introduced in Section 3.6.

```
> EQN := a*x^2 + b*x + c = 0:
> ROOTS := [ solve( EQN, x ) ]:
> R := unapply( ROOTS, (a,b,c) );
```

$$R := (a, b, c) \rightarrow \left[\frac{1}{2}\,\frac{-b + \sqrt{b^2 - 4\,a\,c}}{a}, \frac{1}{2}\,\frac{-b - \sqrt{b^2 - 4\,a\,c}}{a}\right]$$

```
> exact := R(1,-3,-1);
```

$$exact := \left[\frac{3}{2} + \frac{1}{2}\sqrt{13}, \frac{3}{2} - \frac{1}{2}\sqrt{13}\right]$$

```
> default := evalf( exact );
```

$$default := [3.302775638, -.302775638]$$

```
> three := evalf( exact, 3 );
```

$$three := [3.31, -.31]$$

```
> round3 := evalf( default, 3 );
```

$$round3 := [3.30, -.303]$$

```
>
```

To evaluate the accuracy of these results we will insert the different values into the left-hand side of the equation. In theory, the results should all be zero.

```
> EQN1 := subs( a=1, b=-3, c=-1, lhs(EQN) );
```

$$EQN1 := x^2 - 3\,x - 1$$

```
>
> seq( simplify(EQN1), x=exact );
```

$$0, 0$$

Good! This confirms that these results are, in fact, exact.

```
>
> seq( EQN1, x=default );
```

$$.6\;10^{-8}, .1\;10^{-8}$$

This is typical of floating-point computations performed with 10 significant digits.

```
>
> seq( EQN1, x=three );
```

$$.0261, .0261$$

Approximating the exact roots using three-digit floating-point arithmetic results in roots that are accurate to only one digit.

```
>
> seq( EQN1, x=round3 );
```

$$-.0100, .000809$$

Truncating the (10-digit) approximate roots to three digits produces noticeably different results. One of the roots is accurate to one significant digit, the other to three digits. In general, only one significant digit of accuracy should be expected.

```
>
```

Summary

The only solutions which exactly satisfy the equation are those in exact. The floating-point solutions in default satisfy the equation to about $10^{(-9)}$ - one part in a billion. The solutions in three and round3 are significantly less accurate. Note also that while three and round3 are both computed using three significant digits, they use different data and result in significantly different approximations, i.e., floating-point arithmetic is not commutative. In general, it is preferable to begin floating-point operations with the most accurate information available.

```
>
```

Problems (pp. 74 -- 76)

Problem 1

(a)

Use Maple's solve command to solve the system of equations $.00001\,u + v = 1, -u + v = 0$.

(b)

Rewrite the system with integer coefficients; find the exact (i.e., rational) solution to this system.

Solution

```
> restart
```

(a) The floating-point system and its solution are

```
> SYS1 := { 10.^(-5)*u+v=1, -u+v=0 };
```

$$SYS1 := \{.00001000000000\,u + v = 1, -u + v = 0\}$$

```
> SOL1 := solve( SYS1, {u,v} );
```

$$SOL1 := \{v = .9999900001, u = .9999900001\}$$

```
>
```

(b) The rational system and solution are

```
> SYS2 := { u+10^5*v=10^5, -u+v=0 };
```

$$SYS2 := \{u + 100000\,v = 100000, -u + v = 0\}$$

```
> SOL2 := solve( SYS2, { u, v } );
```

$$SOL2 := \{v = \frac{100000}{100001}, u = \frac{100000}{100001}\}$$

```
>
```

As a final test, convert the exact solution to floating-point numbers:

```
> evalf(SOL2);
```

$$\{v = .9999900001, u = .9999900001\}$$

```
>
```

Problem 2

To verify that the solutions found in Problem 1 are correct, use subs to substitute the solutions back into both systems of equations. Further, substitute the rational solution into the original system and the floating-point solution into the integer system.

Note that some numbers are integers and others are floating-point. There is a difference.

To illustrate, use the evalb command (see ?evalb) to see if Maple thinks the equations are satisfied. Explain the results.

Solution

To check that the solutions are correct:

```
> CHK11 := subs( SOL1, SYS1 );
```

$$CHK11 := \{1.000000000 = 1, 0 = 0\}$$

```
> CHK22 := subs( SOL2, SYS2 );
```

$$CHK22 := \{0 = 0, 100000 = 100000\}$$

```
>
```

And, substituting the solutions into the other form of the system:

```
> CHK12 := subs( SOL1, SYS2 );
```

$$CHK12 := \{0 = 0, 100000.0000 = 100000\}$$

```
> CHK21 := subs( SOL2, SYS1 );
```

$$CHK21 := \{1.000000000 = 1, 0 = 0\}$$

```
>
```

These results might give the appearance that both solutions satisfy either form of the system. Recall, however, that a number and its floating-point representation are not equal. The easiest way to see this is to use the map command (see ?map) to apply evalb to each equation in CHK11, CHK12, CHK21, and CHK22.

```
> map( evalb, CHK11 );
```

$$\{false, true\}$$

```
> map( evalb, CHK12 );
```

$$\{false, true\}$$

```
> map( evalb, CHK21 );
```

$$\{false, true\}$$

```
> map( evalb, CHK22 );
```

$$\{true\}$$

Note that the floating-point solution would ''exactly'' solve the system if the RHS of the first equation were a floating-point 1, i.e., replace 1 with 1. in SYS1.

```
>
```

Problem 3

This problem illustrates some of the difficulties that can occur when subtracting floating-point numbers.

Compute the floating-point approximation to the difference of $N1 = 8721\sqrt{3}$, $N2 = 10681\sqrt{2}$, $SUM = 8721\sqrt{3} + 10681\sqrt{2}$, and $DIFF = 8721\sqrt{3} - 10681\sqrt{2}$ using 2, 3, 4, .., 19, 20 significant digits.

To how many digits do N1 and N2 agree?

What are the values of SUM and DIFF, accurate to five significant digits? How many floating-point digits are needed to compute SUM and DIFF to this accuracy?

A more reliable way to compute the difference is to note that PROD=DIFF*SUM is an integer when fully simplified. (Why?) Thus, DIFF = PROD/SUM which can be computed without any subtraction. How many floating-point digits are needed to obtain five significant digits of accuracy in the value of DIFF when it is computed by division?

One moral of this exercise is that the accuracy of a floating-point calculation may not be the same as the number of significant digits used in a calculation. This is a general property of floating-point arithmetic, not just Maple.

Correction

Delete the phrase "the difference of" that immediately precedes the definition of $N1$.

Solution

```
> restart;
```

First, the definitions of the relevant quantities:

```
> N1 := 8721*sqrt(3);
```

$$N1 := 8721\sqrt{3}$$

```
> N2 := 10681*sqrt(2);
```

$$N2 := 10681\sqrt{2}$$

```
> SUM := N1 + N2;
```

$$SUM := 8721\sqrt{3} + 10681\sqrt{2}$$

```
> DIFF := N1 - N2;
```

$$DIFF := 8721\sqrt{3} - 10681\sqrt{2}$$

```
>
```

The seq command simplifies the computation of the desired floating-point approximations

```
> seq( evalf( N1, d ), d=2..20 );
```

15000., 15100., 15100., 15106., 15105.2, 15105.22, 15105.215, 15105.2151, 15105.21510, 15105.215093, 15105.2150928, 15105.21509281, 15105.215092808, 15105.2150928082, 15105.21509280818, 15105.215092808179, 15105.2150928081788, 15105.21509280817888, 15105.215092808178877

```
> seq( evalf( N2, d ), d=2..20 );
```

15000., 15100., 15100., 15105., 15105.2, 15105.22, 15105.215, 15105.2150, 15105.21506, 15105.215060, 15105.2150597, 15105.21505971, 15105.215059707, 15105.2150597071, 15105.21505970703, 15105.215059707029, 15105.2150597070282, 15105.21505970702822, 15105.215059707028216

```
> seq( evalf( SUM, d ), d=2..20 );
```

30000., 30200., 30200., 30211., 30210.4, 30210.44, 30210.430, 30210.4301, 30210.43016, 30210.430153, 30210.4301525, 30210.43015252, 30210.430152515, 30210.4301525153, 30210.43015251521, 30210.430152515208, 30210.4301525152070, 30210.43015251520710, 30210.430152515207093

```
> seq( evalf( DIFF, d ), d=2..20 );
```

0, 0, 0, 1., 0, 0, 0, .0001, .00004, .000033, .0000331, .00003310, .000033101, .0000331011, .00003310115, .000033101150, .0000331011506, .00003310115066, .000033101150661

Note: alternate solution using for ... do ... od;

A solution that avoids seq and that presents all four values together on the same line can be obtained using Maple's repetition command (see Chapter 7).

```
> for d from 2 to 20 do
>   evalf( [ N1, N2, SUM, DIFF ], d );
> od;
```

[15000., 15000., 30000., 0]

[15100., 15100., 30200., 0]

[15100., 15100., 30200., 0]

[15106., 15105., 30211., 1.]

[15105.2, 15105.2, 30210.4, 0]

[15105.22, 15105.22, 30210.44, 0]

[15105.215, 15105.215, 30210.430, 0]

[15105.2151, 15105.2150, 30210.4301, .0001]

[15105.21510, 15105.21506, 30210.43016, .00004]

[15105.215093, 15105.215060, 30210.430153, .000033]

[15105.2150928, 15105.2150597, 30210.4301525, .0000331]

[15105.21509281, 15105.21505971, 30210.43015252, .00003310]

[15105.215092808, 15105.215059707, 30210.430152515, .000033101]

[15105.2150928082, 15105.2150597071, 30210.4301525153, .0000331011]

[15105.21509280818, 15105.21505970703, 30210.43015251521, .00003310115]

[15105.215092808179, 15105.215059707029, 30210.430152515208, .000033101150]

[15105.2150928081788, 15105.2150597070282, 30210.4301525152070, .0000331011506]

[15105.21509280817888, 15105.21505970702822, 30210.43015251520710, .00003310115066]

[15105.215092808178877, 15105.215059707028216, 30210.430152515207093, .000033101150661]

```
>
```

The corresponding command using seq is valid, but the output is not one list per line.

```
> seq( evalf( [ N1, N2, SUM, DIFF ], d ), d=2..20 );
>
```

It is easily seen that N1 and N2 differ in the fifth digit to the right of the decimal point; they are equal for ten significant digits.

>

A close examination of the earlier results indicates that $SUM = 30210$ and $DIFF = .000033101$ to five significant digits. The value of SUM is obtained using 6 significant digits; the value of DIFF requires 14 significant digits.

>

```
> PROD := expand( SUM*DIFF );
```

$$PROD := 1$$

```
> DIFF2 := PROD/SUM;
```

$$DIFF2 := \frac{1}{8721\sqrt{3} + 10681\sqrt{2}}$$

```
> seq( evalf( DIFF2, d ), d=2..20 );
```

.000033, .0000331, .00003311, .000033101, .0000331012, .00003310114, .000033101151, .0000331011507, .00003310115065, .000033101150660, .0000331011506606, .00003310115066060, .000033101150660602, .0000331011506606019, .00003310115066060202, .000033101150660602021, .0000331011506606020224, .00003310115066060202227, .000033101150660602022281

In this way the value of DIFF, accurate to five significant digits, is obtained using only five significant digits -- quite an improvement over the direct approach!

>

Problem 4

Use subs to verify that both solutions found in Example 3-11 are, in fact, points of intersection of the two curves. In general, there are two solutions. Find values of r for which there are no solutions and a single solution. Can there ever be three points of intersection?

Solution

```
> restart;
```

Recall the definitions made in the solution to Example 3-11 (p. 54).

```
> LINE := x + y = 1:
> CIRCLE := x^2 + y^2 = r^2:
> SYS := { LINE, CIRCLE }:
> VARS := { x, y }:
> SOL := solve( SYS, VARS ):
> SOL := [ allvalues( SOL ) ];
```

$$SOL := \left[\{ y = \frac{1}{2} + \frac{1}{2}\sqrt{-1+2\,r^2},\ x = \frac{1}{2} - \frac{1}{2}\sqrt{-1+2\,r^2} \},\ \{ y = \frac{1}{2} - \frac{1}{2}\sqrt{-1+2\,r^2},\ x = \frac{1}{2} + \frac{1}{2}\sqrt{-1+2\,r^2} \} \right]$$

>

To verify that the first solution actually satisfies both equations, substitute the solution back into the two equations.

```
> subs( SOL[1], SYS );
```

$$\{ 1 = 1, \left(\frac{1}{2} - \frac{1}{2}\sqrt{-1+2\,r^2}\right)^2 + \left(\frac{1}{2} + \frac{1}{2}\sqrt{-1+2\,r^2}\right)^2 = r^2 \}$$

Then, simplify the expressions.

```
> simplify( " );
```

$$\{ 1 = 1, r^2 = r^2 \}$$

It is now easily seen that this solution does, in fact, satisfy both equations.

>

A streamlined approach is demostrated for the second solution:

```
> simplify( subs( SOL[2], SYS ) );
```

$$\{ 1 = 1, r^2 = r^2 \}$$

>

The system has exactly one solution when the radicand in SOL is zero, i.e., when $2\,r^2 = 1$:

```
> solve( subs( r=sqrt(1/2), SYS ), VARS );
```

$$\{y = \frac{1}{2}, x = \frac{1}{2}\}, \{y = \frac{1}{2}, x = \frac{1}{2}\}$$

And, there is no (real-valued) solution when the radicand in SOL is negative, i.e., when $2r^2 < 1$:

```
> solve( subs( r=1/2, SYS ), VARS );
> allvalues( " );
```

$$\{x = -\frac{1}{2}\text{RootOf}(2\_Z^2 - 4\_Z + 3) + 1, y = \frac{1}{2}\text{RootOf}(2\_Z^2 - 4\_Z + 3)\}$$

$$\{x = \frac{1}{2} - \frac{1}{4} I\sqrt{2}, y = \frac{1}{2} + \frac{1}{4} I\sqrt{2}\}, \{x = \frac{1}{2} + \frac{1}{4} I\sqrt{2}, y = \frac{1}{2} - \frac{1}{4} I\sqrt{2}\}$$

```
>
```

Problem 5

Calculate the speed of sound in air at sea level and at 35,000 feet (in m/s and in ft/sec) using the formulas provided in the text.

Solution

```
> restart;
```

We begin with the speed of sound at sea level, taking care to convert inches to feet in P_{SL} . (Note that gamma is a protected name in Maple, in order to use this name in our calculations, we have to explicitly tell Maple to remove its protection of this name with the unprotect command.)

```
> P[SL] := 14.7*12^2;
```

$$P_{SL} := 2116.8$$

```
> rho[SL] := 0.002378;
```

$$\rho_{SL} := .002378$$

```
> unprotect(gamma);
> gamma := 1.4;
```

$$\gamma := 1.4$$

The speed of sound is thus found to be (in feet per second)

```
> a[SL] := sqrt( gamma*P[SL]/rho[SL] );
```

$$a_{SL} := 1116.343906$$

The speed of sound at 35000 feet is computed similarly. The pressure and density at this altitude are all that are needed

```
> delta[35000] := 0.2351;
```

$$\delta_{35000} := .2351$$

```
> sigma[35000] := 0.3096;
```

$$\sigma_{35000} := .3096$$

```
> P[35000] := delta[35000]*P[SL];
```

$$P_{35000} := 497.65968$$

```
> rho[35000] := sigma[35000]*rho[SL];
```

$$\rho_{35000} := .0007362288$$

```
> a[35000] := sqrt( gamma*P[35000]/rho[35000] );
```

$$a_{35000} := 972.8006333$$

Thus, the speed of sound slows a little more than 10% at an altitude of 35000 feet compared to sea level. To be more precise

```
> (a[35000]-a[SL])/a[SL];
```

$$-.1285833800$$

```
>
```

To convert from ft/sec to m/s, note that there are 12 inches per foot, 2.54 centimeter per inch, and 0.01 meters per centimeter.

```
> ft2m := 0.01 * 2.54 * 12;
```

$$ft2m := .3048$$

Thus there are 0.3048 m/ft. (Use your common sense to check that this answer is reasonable!)

In m/s, the speed of sound at sea level is

```
> a[SL]*ft2m;
```

$$340.2616225$$

and at 35000 ft it is

```
> a[35000]*ft2m;
```

$$296.5096330$$

```
>
```

Problem 6

It is clear that the weight of an airplane decreases as fuel is consumed. Therefore, the lift required to maintain the cruising altitude will decrease as fuel is consumed. This has not been taken into account in the application. Given a particular fuel consumption rate and starting weight, the distance, s, a plane can travel is given by:

```
> s:=(V/(TSFC*g))*(ln(m[0]/m))*[L/D];
```

$$s := \frac{V \ln\left(\frac{m_0}{m}\right)\left[\frac{L}{D}\right]}{TSFC\, g}$$

where $TSFC$ is the Thrust-Specific Fuel Consumption, g is the gravitational acceleration (32.1740 $\frac{ft}{\sec^2}$), m_0 is the initial mass, and m is the final mass.

Assuming that TSFC=0.75 lb_m/lb_f-hr and that the maximum fuel capacity is 180,000 lb, determine the maximum range based on the lift and drag results required for level flight at 35,000 feet. Determine the minimum amount of fuel required for this aircraft to fly across the United States (approximate distance of 3500 miles).

(The preceding formula was derived by Breguet. This derivation of this equation, which involves differential equations, will be explored in more detail in Problem 13 in Chapter 6.)

Hint

$\frac{m_0}{m}$=1.56. Watch the units; TSFC has hours, not seconds.)

Solution

```
> restart;
```

The information from the Application that is needed for this problem consists of the definitions of lift and drag

```
> lift := rho*V^2/2 * S * CL:
> drag := rho*V^2/2 * S * CD:
> LandD := [ L=lift, D=drag ];
```

$$LandD := \left[L = \frac{1}{2}\rho\, V^2\, S\, CL,\, D = \frac{1}{2}\rho\, V^2\, S\, CD\right]$$

and a few parameter values associated with level flight at 35,000 feet and Mach 0.84

```
> param35 := V=M*a, M=0.84, a=972.8, CL=0.5066, CD=0.0297;
```

$$param35 := V = M\, a,\, M = .84,\, a = 972.8,\, CL = .5066,\, CD = .0297$$

```
>
```

The weight of the plane is the sum of the weight of the fuel, m_F, and the weight of the airplane (and passengers and/or cargo), m_E.

```
> paramFUEL := m0 = mE+mF, m = mE, mE = 500000-mF;
```

$$paramFUEL := m0 = mE + mF,\, m = mE,\, mE = 500000 - mF$$

```
>
```

The range of the plane is given by the formula

```
> s:=(V/(TSFC*g))*(ln(m0/m))*L/D;
```

$$s := \frac{V \ln\left(\frac{m0}{m}\right) L}{TSFC\ g\ D}$$

where the Thurst-Specific Fuel Consumption (with units converted as discussed on pp. 61 -- 62) is

```
> paramTSFC := TSFC = 0.75/g/secperhr, g=32.174, secperhr=60^2;
```

$$paramTSFC := TSFC = \frac{.75}{g\ secperhr},\ g = 32.174,\ secperhr = 3600$$

When all of these values are substituted into the equation for the range of the airplane, we obtain the range in terms of the fuel weight.

```
> RANGE := subs( LandD, param35, paramFUEL, paramTSFC, s );
```

$$RANGE := .6690411364\ 10^8 \ln\left(\frac{500000}{500000 - mF}\right)$$

Note that this range has the units of <u>feet</u> - to convert to miles, divide by 5280 (feet per mile).

```
> subs( mF=180000, RANGE );
```

$$.6690411364\ 10^8 \ln\left(\frac{25}{16}\right)$$

```
> evalf( "/5280 );
```

$$5655.008148$$

Thus, this airplane has a cruising range of approximately 5655 miles.

```
>
```

The minimum amount of fuel (in pounds) needed to fly across the United States (3500 miles) is

```
> solve( RANGE=3500*5280, { mF } );
```

$$\{ mF = 120675.5532 \}$$

```
>
```

Problem 7

Express the thrust needed to keep an aircraft at cruising altitude in terms of the aircraft's weight, aspect ratio, wing span, and Mach number when altitude is 35,000 feet and the lift-to-drag coefficients are (C_{DO},α)=(0.0155, 0.0588). As an aeronautical engineer, explain what changes in the aircraft's weight, wing span, aspect ratio, and Mach number would decrease the thrust requirement.

Solution

```
> restart;
```

The first step in obtaining the desired expression for the thrust is to recall the two balance laws, the definitions of lift and drag, and the lift-to-drag relationship from the Application:

```
> lift := rho*V^2/2 * S * CL:
> drag := rho*V^2/2 * S * CD:
> balance1 := lift=weight;
> balance2 := thrust=drag;
```

$$balance1 := \frac{1}{2} \rho V^2 S\ CL = weight$$

$$balance2 := thrust = \frac{1}{2} \rho V^2 S\ CD$$

```
> liftdrag := CD=CD0 + alpha*CL^2;
```

$$liftdrag := CD = CD0 + \alpha\ CL^2$$

```
> T1 := subs( liftdrag, solve(balance1,{CL}), balance2 );
```

$$T1 := thrust = \frac{1}{2} \rho V^2 S \left(CD0 + 4 \frac{\alpha\ weight^2}{\rho^2 V^4 S^2} \right)$$

```
>
```

The variables V, S, and rho all need to be replaced with equivalent expressions in terms of the desired quantities (and known physical constants).

```
> param := V=M*a, S=b^2/AR, rho=delta*rho0;
```

$$param := V = M\,a,\ S = \frac{b^2}{AR},\ \rho = \delta\,\rho 0$$

```
> param0 := p0=14.696*12^2, rho0=0.002377, gamma1=1.4;
```

$$param0 := p0 = 2116.224,\ \rho 0 = .002377,\ \gamma 1 = 1.4$$

```
> param35 := delta=0.2315, sigma=0.3096, a=972.8;
```

$$param35 := \delta = .2315,\ \sigma = .3096,\ a = 972.8$$

```
> T2 := subs( param, param0, param35, T1 );
```

$$T2 := thrust = 260.3738143\,\frac{M^2\,b^2\left(CD0 + .00001475045404\,\frac{\alpha\,weight^2\,AR^2}{M^4\,b^4}\right)}{AR}$$

The derivation is completed by using the given values for the lift-to-drag conversion (and converting the result into a function)

```
> paramLD := CD0=0.0155, alpha=0.0588;
```

$$paramLD := CD0 = .0155,\ \alpha = .0588$$

```
> T3 := subs( paramLD, T2 );
```

$$T3 := thrust = 260.3738143\,\frac{M^2\,b^2\left(.0155 + .8673266976\;10^{-6}\,\frac{weight^2\,AR^2}{M^4\,b^4}\right)}{AR}$$

```
> THRUST := unapply( rhs(T3), (weight,AR,b,M) );
```

$$THRUST := (weight, AR, b, M) \rightarrow 260.3738143\,\frac{M^2\,b^2\left(.0155 + .8673266976\;10^{-6}\,\frac{weight^2\,AR^2}{M^4\,b^4}\right)}{AR}$$

```
>
```

For the airplane discussed in the Application, with the parameter values given in this problem, the necessary thrust (in pounds) is

```
> THRUST( 500000, 10, 200, 0.84 );
```

$$31393.91691$$

The simplest observation is that as the airplane's weight increases, the thrust increases. The dependence on M, b, and AR is more subtle. Note that if we define $\omega = \frac{(M\,b)^2}{AR}$, then the thrust depends only on ω and the weight.

```
> THRUST( weight, (M*b)^2/omega, b, M );
```

$$260.3738143\,\omega\left(.0155 + .8673266976\;10^{-6}\,\frac{weight^2}{\omega^2}\right)$$

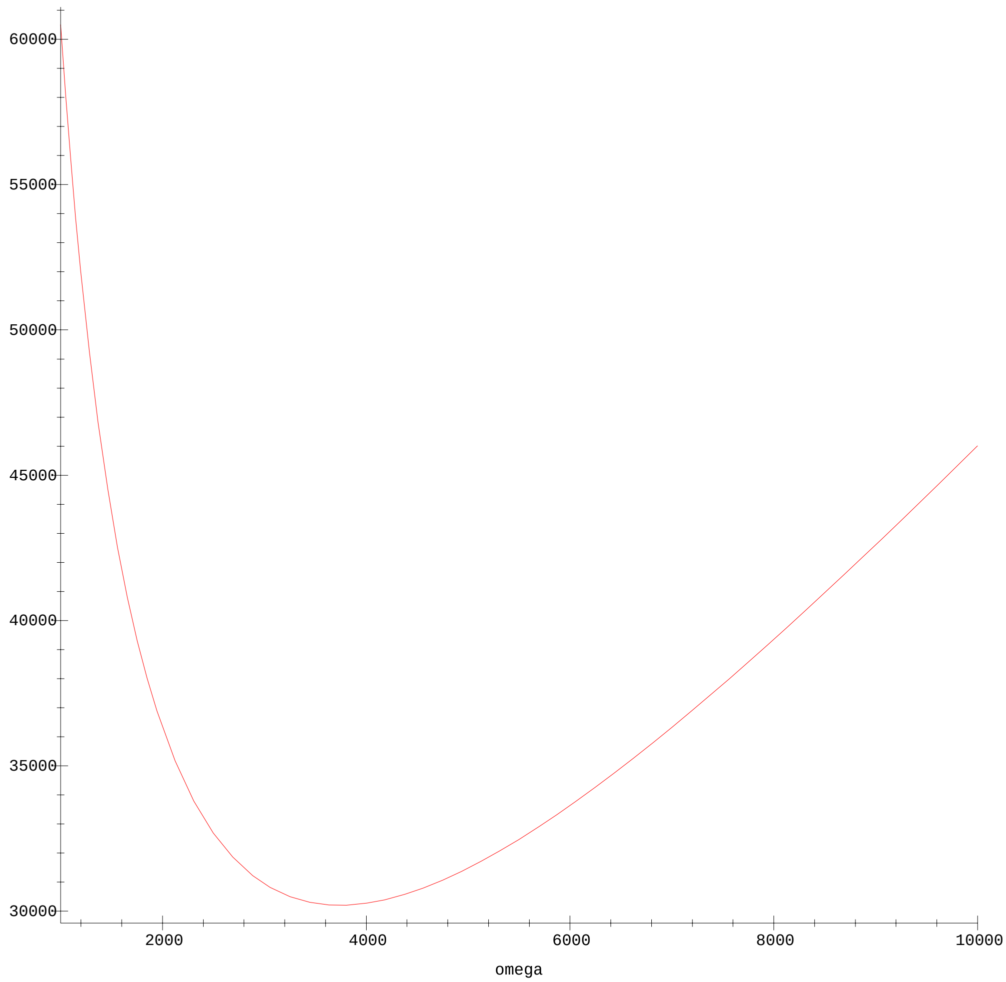
Do not be misled by the small coefficient. Since the weight is of the order of 10^5, that term can be quite large

```
> THRUST( 500000, (M*b)^2/omega, b, M );
```

$$260.3738143\,\omega\left(.0155 + \frac{216831.6744}{\omega^2}\right)$$

A plot is the simplest way to understand the dependence of thrust on ω:

```
> plot( THRUST( 500000, (M*b)^2/omega, b, M ), omega=1000..10000 );
```



>

With the parameters given for this airplane, and at Mach 0.84, ω=2822.4. Thus, the thrust can be decreased by changing M, b, and AR so that ω increases - but does not exceed approximately 3740. Or, recalling that $AR = \frac{b^2}{S}$, we could state the condition as $S M^2$ <= 3700.

>

Problem 8

Determine the range of an airplane at cruising altitude in terms of its "empty" weight (that is, no passengers and no fuel), and in terms of the amount of fuel, wing span, Mach number, aspect ratio, $TSFC$, α, γ, δ, and σ. Give your answer in miles. Determine whether cruising 3,000 feet above or below the 35,000 foot cruising altitude increases the aircraft's range. Use δ_{32000}=0.2707, δ_{38000}=0.2037, σ_{32000}= 0.3471, and σ_{38000}=0.2692.

Clarification

The "empty" weight should include the weight of the passengers (or other cargo). If not, then this weight would need to be another variable in the problem. Also, in addition to the parameters listed in the problem statement, the expression for the range will include universal constants (g, γ, ρ_{SL}, p_{SL}) and C_{D_0} .

Solution

```
> restart;
```

This problem begins exactly like Problem 7. The original expression for the range - in miles - is

```
> s:=(V/(TSFC*g))*(ln(m0/m))*L/D/5280;
```

$$s := \frac{1}{5280}\,\frac{V \ln\left(\frac{m0}{m}\right) L}{TSFC\, g\, D}$$

```
>
```

To express the range in terms of the stated parameters it is necessary to collect various relationships between lift, drag, and the coefficients of lift and drag:

```
> lift := rho*V^2/2 * S * CL:
> drag := rho*V^2/2 * S * CD:
> paramLD := L=lift, D=drag;
```

$$paramLD := L = \frac{1}{2}\,\rho\, V^2\, S\, CL,\ D = \frac{1}{2}\,\rho\, V^2\, S\, CD$$

```
> paramLD2 := CD=CD0 + alpha*CL^2, CL=2*m/rho/V^2/S;
```

$$paramLD2 := CD = CD0 + \alpha\, CL^2,\ CL = 2\,\frac{m}{\rho\, V^2\, S}$$

This information can be used to express the range as

```
> s1 := subs( paramLD, paramLD2, s );
```

$$s1 := \frac{1}{2640}\,\frac{\ln\left(\frac{m0}{m}\right) m}{V\, TSFC\, g\, \rho\, S \left(CD0 + 4\,\frac{\alpha\, m^2}{\rho^2\, V^4\, S^2}\right)}$$

```
>
```

The airplane’s weights are expressed in terms of the "empty" and "full" weights as in Problem 6.

```
> paramW := m0=mE+mF, m=mE;
```

$$paramW := m0 = mE + mF,\ m = mE$$

Other relationships between the physical and dimensionless parameters are also needed:

```
> param := S=b^2/AR, V=M*a, a = sqrt(gamma1*p/rho), rho=sigma*rho0, p=delta*p0;
```

$$param := S = \frac{b^2}{AR},\ V = M\, a,\ a = \sqrt{\frac{\gamma 1\, p}{\rho}},\ \rho = \sigma\, \rho 0,\ p = \delta\, p0$$

```
> RANGE := subs( paramW, param, s1 );
```

$$RANGE := \frac{1}{2640}\,\frac{\ln\left(\frac{mE + mF}{mE}\right) mE\, AR}{M \sqrt{\frac{\gamma 1\, \delta\, p0}{\sigma\, \rho 0}}\, TSFC\, g\, \sigma\, \rho 0\, b^2 \left(CD0 + 4\,\frac{\alpha\, mE^2\, AR^2}{M^4\, \gamma 1^2\, \delta^2\, p0^2\, b^4}\right)}$$

```
>
```

To determine whether the range is greater at a cruising altitude of 32000 feet or 38000 feet, the relative air pressure and density are required for each altitude:

```
> param35 := delta=0.2315, sigma=0.3096;
> param32 := delta=0.2707, sigma=0.3471;
> param38 := delta=0.2037, sigma=0.2692;
```

$$param35 := \delta = .2315,\ \sigma = .3096$$

$$param32 := \delta = .2707,\ \sigma = .3471$$

$$param38 := \delta = .2037,\ \sigma = .2692$$

The corresponding ranges are:

```
> range35 := subs( param35, RANGE );
```

$$range35 := .001414881602 \frac{\ln\left(\frac{mE+mF}{mE}\right) mE\, AR}{M \sqrt{\frac{\gamma 1\, p0}{\rho 0}}\, TSFC\, g\, \rho 0\, b^2 \left(CD0 + 74.63765752 \frac{\alpha\, mE^2\, AR^2}{M^4\, \gamma 1^2\, p0^2\, b^4}\right)}$$

```
> range32 := subs( param32, RANGE );
```

$$range32 := .001235732982 \frac{\ln\left(\frac{mE+mF}{mE}\right) mE\, AR}{M \sqrt{\frac{\gamma 1\, p0}{\rho 0}}\, TSFC\, g\, \rho 0\, b^2 \left(CD0 + 54.58627764 \frac{\alpha\, mE^2\, AR^2}{M^4\, \gamma 1^2\, p0^2\, b^4}\right)}$$

```
> range38 := subs( param38, RANGE );
```

$$range38 := .001617569803 \frac{\ln\left(\frac{mE+mF}{mE}\right) mE\, AR}{M \sqrt{\frac{\gamma 1\, p0}{\rho 0}}\, TSFC\, g\, \rho 0\, b^2 \left(CD0 + 96.40019964 \frac{\alpha\, mE^2\, AR^2}{M^4\, \gamma 1^2\, p0^2\, b^4}\right)}$$

While these expressions for the ranges are quite complicated, the ratio with the range at 35000 feet can be used to answer this question

```
> RATIO38 := range38/range35;
```

$$RATIO38 := 1.143254531 \frac{CD0 + 74.63765752 \frac{\alpha\, mE^2\, AR^2}{M^4\, \gamma 1^2\, p0^2\, b^4}}{CD0 + 96.40019964 \frac{\alpha\, mE^2\, AR^2}{M^4\, \gamma 1^2\, p0^2\, b^4}}$$

```
> RATIO32 := range32/range35;
```

$$RATIO32 := .8733826069 \frac{CD0 + 74.63765752 \frac{\alpha\, mE^2\, AR^2}{M^4\, \gamma 1^2\, p0^2\, b^4}}{CD0 + 54.58627764 \frac{\alpha\, mE^2\, AR^2}{M^4\, \gamma 1^2\, p0^2\, b^4}}$$

The size of the rational expression that appears in the numerator and denominator of each ratio is critical to the analysis. Fortunately, most of the parameters involved in this expression are well-known for this problem:

```
> param0 := p0=14.696*12^2, gamma1=1.4, b=200, AR=10, M=0.84, mE=320000;
```

$$param0 := p0 = 2116.224,\ \gamma 1 = 1.4,\ b = 200,\ AR = 10,\ M = .84,\ mE = 320000$$

```
> subs( param0, RATIO38 );
```

$$1.143254531 \frac{CD0 + .1093053672\, \alpha}{CD0 + .1411761781\, \alpha}$$

```
> subs( param0, RATIO32 );
```

$$.8733826069 \frac{CD0 + .1093053672\, \alpha}{CD0 + .07994051957\, \alpha}$$

While the values of C_{D_0} and α depend on the how the data is used, recall that both of these parameters are of the same magnitude. Thus, since the rational expressions in the ratios of the ranges are essentially 1, the range is increased (by less than 14%) when the cruising altitude is 38000 feet and is decreased (by less than 12%) when the cruising altitude is 32000 feet. (This answer begins to change as the ratio $\frac{\alpha}{C_{D_0}}$ exceeds about 10.)

```
>
```

Problem 9

Fuzzy sets are used in an increasing number of engineering disciplines to more accurately mimic the manner in which human beings make decisions. This general area of study is often referred to as fuzzy logic. For example, a fuzzy logic decision-making circuitry could be incorporated into the timing mechanism of a dishwasher to determine to what extent the dishes within are clean or dirty, or, more important, partially dirty. In this fashion, the dishwasher could be made to operate more efficiently if fuzzy logic can be used to shut off the dishwasher as soon as the dishes are determined to be clean instead of simply running for a fixed amount of time.

An example of a fuzzy set is "numbers close to 10". A traditional "crisp" set would, for example, give a value of 1 for all numbers between 8 and 12 to indicate full memberhsip in this set. All numbers outside this range would get a value 0, to indicate nonmembership. But this is not realistic, since the number 7.9 should have greater membership status than the number 2.5. Suppose, instead, we use a fuzzy set and describe numbers close to 10 by the membership function

$$\mu(x) = \frac{1}{1+(x-10)^2}$$

Note that $x=10$ has full membership status, since $\mu(10)=1$, and all other values of x have partial membership status, with values close to 10 having a status closer to full membership. In fuzzy logic, nonmembership in a fuzzy set A is determined by the membership function for $C=\overline{A}$, the complement of A; that is, if μ_A is the membership function for A, then the membership function for C is $\mu_C = 1-\mu_A$. (Note that when $A=\{10\}$, $\mu_c(10)$=1-$\mu_A(10)$=0 so that the number 10 has nonmembership in C while all other values of x have partial membership in C.)

(a) Define and plot the membership functions for $A=\{10\}$ and the complement of A.
(b) Define and plot the membership function for $B=\{15\}$.
(c) Determine a membership function for the union of A and B, that is, for the numbers close to 10 OR close to 15.

Hint

What properties should this function have? Plot the membership function for A and for B on the same axis. How can these functions be combined to create a function with the necessary properties? See ?max.)

(d) Determine a membership function for the intersection of A and B, that is, for the numbers close to 10 AND close to 15.

Hint

This membership function never takes on the value 1 since no element has full membership in both A and B.)

Correction

The description of the set C is correct, however, the definition should be $C=\overline{A}$.

Solution

```
> restart;
```

This problem can be solved using either functions or expressions. The expressions are probably easier to use for plotting, but functions are advantageous for many other situations. Both versions are provided here; a more efficient implementation using functions is discussed in Chapter 6.

Solution 1: expressions

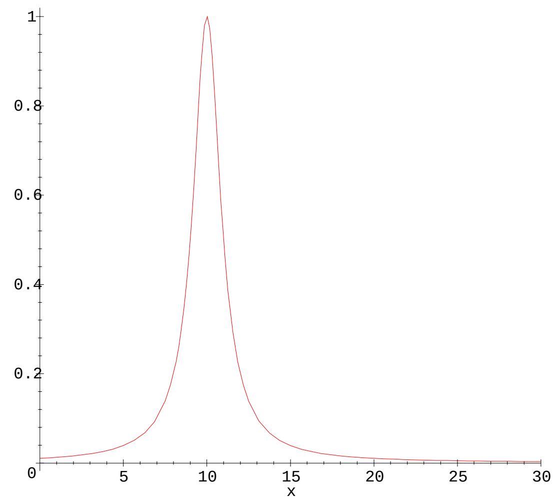
(a) The membership function for $A=\{10\}$ is represented by the expression

```
> mu[A] := 1/(1+(x-10)^2);
```

$$\mu_A := \frac{1}{1+(x-10)^2}$$

```
> plot( mu[A], x=0..30, title=‘Membership function for A={10}‘ );
```

Membership function for A={10}



Observe that the membership function is defined for all real numbers, has values between 0 and 1, and the value 1 occurs only for elements of the set A, i.e., $x = 10$.

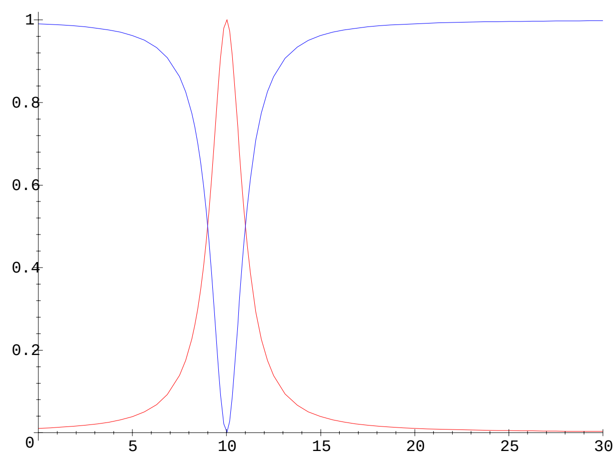
The membership function for the complement set would be

```
> mu[notA] := 1 - mu[A];
```

$$\mu_{notA} := 1 - \frac{1}{1 + (x - 10)^2}$$

```
> plot( [mu[A], mu[notA]], x=0..30, color=[red,blue], title=‘Membership
  functions for A and complement(A)‘ );
```

Membership functions for A and complement(A)



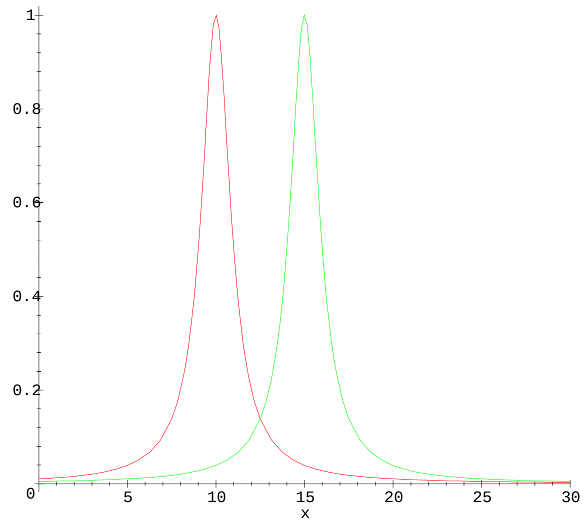
(b) The membership function for $B = \{15\}$ is

```
> mu[B] := 1/(1+(x-15)^2);
```

$$\mu_B := \frac{1}{1 + (x - 15)^2}$$

```
> plot( [mu[A], mu[B]], x=0..30, color=[red,green], title=‘Membership
  functions for A and B‘ );
```

Membership functions for A and B



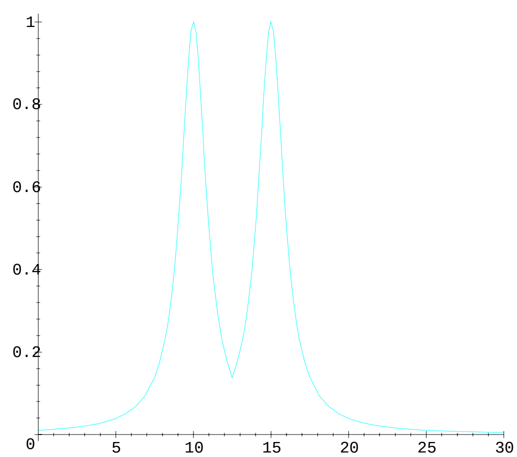
(c) This membership function should be 1 for both $x = 10$ and $x = 15$ and numbers equally close to 10 and to 15 (e.g., 9 and 11 and 14 and 16) should all have the same membership value. The simplest function that satisfies these properties is the maximum of the two membership functions

```
> mu[AorB] := max(mu[A],mu[B]);
```

$$\mu_{AorB} := \max\left(\frac{1}{1+(x-10)^2}, \frac{1}{1+(x-15)^2}\right)$$

```
> plot( [mu[AorB]], x=0..30, color=[cyan], title=‘Membership functions for "A or B"‘ );
```

Membership functions for "A or B"



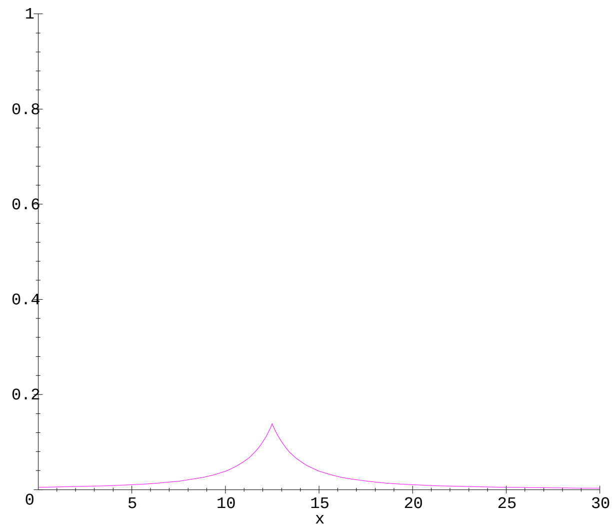
(d) Based on the discussion in (c), it seems the minimum of the membership functions for A and for B should be an appropriate membership function for "A and B".

```
> mu[AandB] := min(mu[A],mu[B]);
```

$$\mu_{AandB} := \min\left(\frac{1}{1+(x-10)^2}, \frac{1}{1+(x-15)^2}\right)$$

```
> plot( mu[AandB], x=0..30, 0..1, color=magenta, title=‘Membership function for "A and B"‘ );
```

Membership function for "A and B"



The fact that this function has a maximum at 12.5 seems appropriate -- this is the number that is closest to being a member of both A and B.

Solution 2: functions

(a)

```
> Mu[A] := x -> 1/(1+(x-10)^2);
```

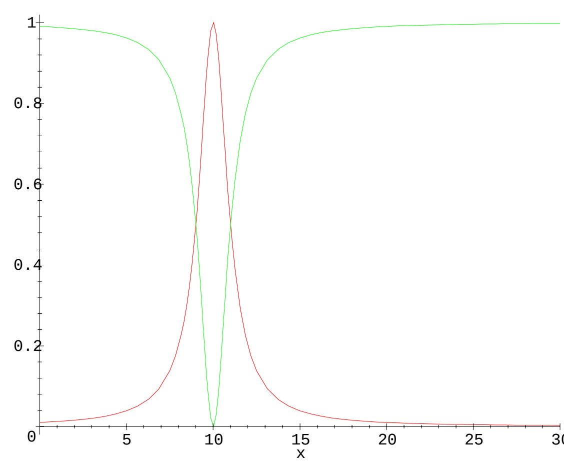
$$M_A := x \rightarrow \frac{1}{1+(x-10)^2}$$

```
> Mu[notA] := 1-Mu[A];
```

$$M_{notA} := 1 - M_A$$

```
> plot( [ Mu[A](x), Mu[notA](x) ], x=0..30, title=`Fuzzy membership in A and
  in complement(A)` );
```

Fuzzy membership in A and in complement(A)



(b)

```
> Mu[B] := x -> 1/(1+(x-15)^2);
```

$$M_B := x \rightarrow \frac{1}{1+(x-15)^2}$$

(c) and (d) Note the specific inclusion of the arguments in the function definitions.

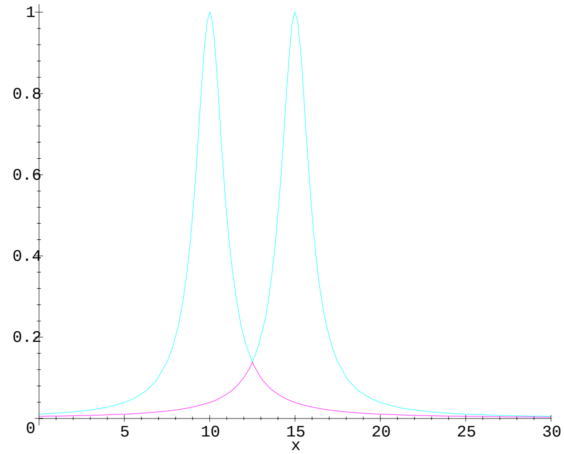
```
> Mu[AorB] := x -> max(Mu[A](x),Mu[B](x));
> Mu[AandB] := x -> min(Mu[A](x),Mu[B](x));
```

$$M_{AorB} := x \to \max(M_A(x), M_B(x))$$

$$M_{AandB} := x \to \min(M_A(x), M_B(x))$$

```
> plot([ Mu[AorB](x), Mu[AandB](x)], x=0..30, color=[cyan,magenta],
  title=`Fuzzy membership in "A or B" and "A and B"` );
```

Fuzzy membership in "A or B" and "A and B"



A more explicit and somewhat more efficient means of implementing the membership functions as functions will be discussed in Chapter 6.

```
>
```

Problem 10

This is a continuation of the Try It! exercise first discussed at the end of Section 3-2.
The exact division points for the 128 levels are not convenient for human analysis; the floating-point form of these levels are almost equally difficult to use. What is needed is a list of the levels with only a few decimal places. Create the list of levels with exactly two decimal digits of accuracy.

Solution

```
> restart;
```

Recall from the Try It! on p. 52 that the exact division levels are

```
> QUANT := [ seq( 50*(i/128), i=0..128 ) ];
```

$$\begin{aligned}QUANT := \Big[&0, \tfrac{25}{64}, \tfrac{25}{32}, \tfrac{75}{64}, \tfrac{25}{16}, \tfrac{125}{64}, \tfrac{75}{32}, \tfrac{175}{64}, \tfrac{25}{8}, \tfrac{225}{64}, \tfrac{125}{32}, \tfrac{275}{64}, \tfrac{75}{16}, \tfrac{325}{64}, \tfrac{175}{32}, \tfrac{375}{64}, \tfrac{25}{4}, \tfrac{425}{64}, \tfrac{225}{32}, \tfrac{475}{64}, \tfrac{125}{16}, \tfrac{525}{64}, \tfrac{275}{32},\\ &\tfrac{575}{64}, \tfrac{75}{8}, \tfrac{625}{64}, \tfrac{325}{32}, \tfrac{675}{64}, \tfrac{175}{16}, \tfrac{725}{64}, \tfrac{375}{32}, \tfrac{775}{64}, \tfrac{25}{2}, \tfrac{825}{64}, \tfrac{425}{32}, \tfrac{875}{64}, \tfrac{225}{16}, \tfrac{925}{64}, \tfrac{475}{32}, \tfrac{975}{64}, \tfrac{125}{8}, \tfrac{1025}{64}, \tfrac{525}{32}, \tfrac{1075}{64}, \tfrac{275}{16}, \tfrac{1125}{64}\\ &\tfrac{575}{32}, \tfrac{1175}{64}, \tfrac{75}{4}, \tfrac{1225}{64}, \tfrac{625}{32}, \tfrac{1275}{64}, \tfrac{325}{16}, \tfrac{1325}{64}, \tfrac{675}{32}, \tfrac{1375}{64}, \tfrac{175}{8}, \tfrac{1425}{64}, \tfrac{725}{32}, \tfrac{1475}{64}, \tfrac{375}{16}, \tfrac{1525}{64}, \tfrac{775}{32}, \tfrac{1575}{64}, 25, \tfrac{1625}{64}, \tfrac{825}{32},\\ &\tfrac{1675}{64}, \tfrac{425}{16}, \tfrac{1725}{64}, \tfrac{875}{32}, \tfrac{1775}{64}, \tfrac{225}{8}, \tfrac{1825}{64}, \tfrac{925}{32}, \tfrac{1875}{64}, \tfrac{475}{16}, \tfrac{1925}{64}, \tfrac{975}{32}, \tfrac{1975}{64}, \tfrac{125}{4}, \tfrac{2025}{64}, \tfrac{1025}{32}, \tfrac{2075}{64}, \tfrac{525}{16}, \tfrac{2125}{64}, \tfrac{1075}{32},\\ &\tfrac{2175}{64}, \tfrac{275}{8}, \tfrac{2225}{64}, \tfrac{1125}{32}, \tfrac{2275}{64}, \tfrac{575}{16}, \tfrac{2325}{64}, \tfrac{1175}{32}, \tfrac{2375}{64}, \tfrac{75}{2}, \tfrac{2425}{64}, \tfrac{1225}{32}, \tfrac{2475}{64}, \tfrac{625}{16}, \tfrac{2525}{64}, \tfrac{1275}{32}, \tfrac{2575}{64}, \tfrac{325}{8}, \tfrac{2625}{64},\\ &\tfrac{1325}{32}, \tfrac{2675}{64}, \tfrac{675}{16}, \tfrac{2725}{64}, \tfrac{1375}{32}, \tfrac{2775}{64}, \tfrac{175}{4}, \tfrac{2825}{64}, \tfrac{1425}{32}, \tfrac{2875}{64}, \tfrac{725}{16}, \tfrac{2925}{64}, \tfrac{1475}{32}, \tfrac{2975}{64}, \tfrac{375}{8}, \tfrac{3025}{64}, \tfrac{1525}{32}, \tfrac{3075}{64}, \tfrac{775}{16},\\ &\tfrac{3125}{64}, \tfrac{1575}{32}, \tfrac{3175}{64}, 50\Big]\end{aligned}$$

Since the largest value is 50, two decimal digits require the use of at least 4 digits in the floating point calculations.

```
> QUANT2 := [ seq( evalf(Q,4), Q=QUANT ) ];
```

$QUANT2 := [0, .3906, .7813, 1.172, 1.563, 1.953, 2.344, 2.734, 3.125, 3.516, 3.906, 4.297, 4.688, 5.078, 5.469,$

5.859, 6.250, 6.641, 7.031, 7.422, 7.813, 8.203, 8.594, 8.984, 9.375, 9.766, 10.16, 10.55, 10.94, 11.33, 11.72, 12.11, 12.50, 12.89, 13.28, 13.67, 14.06, 14.45, 14.84, 15.23, 15.63, 16.02, 16.41, 16.80, 17.19, 17.58, 17.97, 18.36, 18.75, 19.14, 19.53, 19.92, 20.31, 20.70, 21.09, 21.48, 21.88, 22.27, 22.66, 23.05, 23.44, 23.83, 24.22, 24.61, 25., 25.39, 25.78, 26.17, 26.56, 26.95, 27.34, 27.73, 28.13, 28.52, 28.91, 29.30, 29.69, 30.08, 30.47, 30.86, 31.25, 31.64, 32.03, 32.42, 32.81, 33.20, 33.59, 33.98, 34.38, 34.77, 35.16, 35.55, 35.94, 36.33, 36.72, 37.11, 37.50, 37.89, 38.28, 38.67, 39.06, 39.45, 39.84, 40.23, 40.63, 41.02, 41.41, 41.80, 42.19, 42.58, 42.97, 43.36, 43.75, 44.14, 44.53, 44.92, 45.31, 45.70, 46.09, 46.48, 46.88, 47.27, 47.66, 48.05, 48.44, 48.83, 49.22, 49.61, 50.]

Notice that the numbers less than 1 have four decimal places and those between 1 and 10 have three decimal places; the problem asks for exactly two decimal digits of accuracy. One way to fulfill this request is to multiply each entry by 100, convert the result to an integer, then divide by 100. That is,

```
> QUANT3 := [ seq( evalf(trunc(100*Q)/100,4), Q=QUANT2 ) ];
```

QUANT3 := [0, .3900, .7800, 1.170, 1.560, 1.950, 2.340, 2.730, 3.120, 3.510, 3.900, 4.290, 4.680, 5.070, 5.460, 5.850, 6.250, 6.640, 7.030, 7.420, 7.810, 8.200, 8.590, 8.980, 9.370, 9.760, 10.16, 10.55, 10.94, 11.33, 11.72, 12.11, 12.50, 12.89, 13.28, 13.67, 14.06, 14.45, 14.84, 15.23, 15.63, 16.02, 16.41, 16.80, 17.19, 17.58, 17.97, 18.36, 18.75, 19.14, 19.53, 19.92, 20.31, 20.70, 21.09, 21.48, 21.88, 22.27, 22.66, 23.05, 23.44, 23.83, 24.22, 24.61, 25., 25.39, 25.78, 26.17, 26.56, 26.95, 27.34, 27.73, 28.13, 28.52, 28.91, 29.30, 29.69, 30.08, 30.47, 30.86, 31.25, 31.64, 32.03, 32.42, 32.81, 33.20, 33.59, 33.98, 34.38, 34.77, 35.16, 35.55, 35.94, 36.33, 36.72, 37.11, 37.50, 37.89, 38.28, 38.67, 39.06, 39.45, 39.84, 40.23, 40.63, 41.02, 41.41, 41.80, 42.19, 42.58, 42.97, 43.36, 43.75, 44.14, 44.53, 44.92, 45.31, 45.70, 46.09, 46.48, 46.88, 47.27, 47.66, 48.05, 48.44, 48.83, 49.22, 49.61, 50.]

An equivalent solution, working directly from QUANT, would be

```
> QUANT4 := [ seq( evalf(trunc(100.*Q)/100.,4), Q=QUANT ) ];
```

QUANT4 := [0, .3900, .7800, 1.170, 1.560, 1.950, 2.340, 2.730, 3.120, 3.510, 3.900, 4.290, 4.680, 5.070, 5.460, 5.850, 6.250, 6.640, 7.030, 7.420, 7.810, 8.200, 8.590, 8.980, 9.370, 9.760, 10.16, 10.55, 10.94, 11.33, 11.72, 12.11, 12.50, 12.89, 13.28, 13.67, 14.06, 14.45, 14.84, 15.23, 15.63, 16.02, 16.41, 16.80, 17.19, 17.58, 17.97, 18.36, 18.75, 19.14, 19.53, 19.92, 20.31, 20.70, 21.09, 21.48, 21.88, 22.27, 22.66, 23.05, 23.44, 23.83, 24.22, 24.61, 25.00, 25.39, 25.78, 26.17, 26.56, 26.95, 27.34, 27.73, 28.13, 28.52, 28.91, 29.30, 29.69, 30.08, 30.47, 30.86, 31.25, 31.64, 32.03, 32.42, 32.81, 33.20, 33.59, 33.98, 34.38, 34.77, 35.16, 35.55, 35.94, 36.33, 36.72, 37.11, 37.50, 37.89, 38.28, 38.67, 39.06, 39.45, 39.84, 40.23, 40.63, 41.02, 41.41, 41.80, 42.19, 42.58, 42.97, 43.36, 43.75, 44.14, 44.53, 44.92, 45.31, 45.70, 46.09, 46.48, 46.88, 47.27, 47.66, 48.05, 48.44, 48.83, 49.22, 49.61, 50.00]

The observant reader should ask whether the reduced floating-point precision used to compute QUANT impact the final result. A quick check of the elements in QUANT3 and QUANT4 shows that the results are in complete agreement

```
> seq( QUANT3[i]-QUANT4[i], i=1..nops(QUANT4) );
```

0, 0

Despite the agreement in this problem, it is generally preferable to retain as much accuracy as long as possible when working with floating-point data.

Note, however, that if trunc is changed to round in QUANT4, then some of the first few levels are slightly different in the second decimal digit.

```
> QUANT5 := [ seq( evalf(round(100.*Q)/100.,4), Q=QUANT ) ];
```

QUANT5 := [0, .3900, .7800, 1.170, 1.560, 1.950, 2.340, 2.730, 3.130, 3.520, 3.910, 4.300, 4.690, 5.080, 5.470, 5.860, 6.250, 6.640, 7.030, 7.420, 7.810, 8.200, 8.590, 8.980, 9.380, 9.770, 10.16, 10.55, 10.94, 11.33, 11.72, 12.11, 12.50, 12.89, 13.28, 13.67, 14.06, 14.45, 14.84, 15.23, 15.63, 16.02, 16.41, 16.80, 17.19, 17.58, 17.97, 18.36, 18.75, 19.14, 19.53, 19.92, 20.31, 20.70, 21.09, 21.48, 21.88, 22.27, 22.66, 23.05, 23.44, 23.83, 24.22, 24.61, 25.00, 25.39, 25.78, 26.17, 26.56, 26.95, 27.34, 27.73, 28.13, 28.52, 28.91, 29.30, 29.69, 30.08, 30.47, 30.86, 31.25, 31.64, 32.03, 32.42, 32.81, 33.20, 33.59, 33.98, 34.38, 34.77, 35.16, 35.55, 35.94, 36.33, 36.72, 37.11, 37.50, 37.89, 38.28, 38.67, 39.06, 39.45, 39.84, 40.23, 40.63, 41.02, 41.41, 41.80, 42.19, 42.58, 42.97, 43.36, 43.75, 44.14, 44.53, 44.92, 45.31, 45.70, 46.09, 46.48, 46.88, 47.27, 47.66, 48.05, 48.44, 48.83, 49.22, 49.61, 50.00]

```
> seq( QUANT5[i]-QUANT4[i], i=1..nops(QUANT4) );
```

0, 0, 0, 0, 0, 0, 0, 0, .010, .010, .010, .010, .010, .010, .010, .010, 0, 0, 0, 0, 0, 0, 0, 0, .010, .010, 0

```
>
```