Contents

Part I

Introduction to Programming for Engineers

1

MATLAB Basics

» **1.** Remove the command history window from the MATLAB environment and then retrieve it.

To remove it, click the "x" in the upper left hand corner of the command history window. To retrieve it, go to **Desktop** → **Command History**.

» **2.** Resize the command prompt so that it takes up less than half of the total MATLAB environment space.

Click and drag the borders of the Command Prompt to achieve this effect.

» **3.** Change the background of the MATLAB environment to black and the font color to orange.

Go to **File** → **Preferences** → **Colors** (PC) or **Matlab** → **Preferences** → **Colors** (MAC), and change the colors as needed.

» **4.** Change the current directory to any folder other than the current default working directory, and then change it back to the default directory.

Change the folder above the Command Prompt.

» **5.** Type `>> travel` into the command prompt. This program tries to solve the Traveling Salesman Problem.

```
>> travel
```

» **6.** Type `>> filterguitar` into the command prompt. This program simulates the sound of a guitar using mathematical and computational methods.

```
>> filterguitar
```

» **7.** Type `>> lorenz` into the command prompt. The Lorenz Attractor is a mathematical model originally formulated to simulate atmospheric weather patterns. However, it has some surprising results that eventually led to the field of Chaos Theory. This program displays the simulation results for different initial conditions.

```
>> lorenz
```

» **8.** Compute the area of a triangle with base 10 and height 12. Recall that the area of a triangle is half the base times the height.

```
>> 0.5*10*12
```

» **9.** Compute the surface area and volume of a cylinder with radius 5 and height 3.

```
>> 2*pi*5^2 + 2*pi*5*3
```

» **10.** Compute the slope between the points $(3, 4)$ and $(5, 9)$. Recall that the slope between points (x_1, y_1) and (x_2, y_2) is $\frac{y_2-y_1}{x_2-x_1}$.

```
>> (9-4)/(5-3)
```

» **11.** Compute the distance between the points $(3, 4)$ and $(5, 9)$. Recall that the distance between points in two dimensions is $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$.

```
>> sqrt((9-4)^2 + (5-3)^2)
```

» **12.** Use MATLAB's `factorial` function to compute 6!.

```
>> factorial(6)
```

» **13.** A year is considered to be 365 days long. However, a more exact figure is 365.24 days. As a consequence, if we held to the standard 365-day year, we would gradually lose that fraction of the day over time, and seasons and other astronomical events would not occur as expected. A leap year is a year that has an extra day, February 29, to keep the timescale on track. Leap years occur on years that are exactly divisible by 4,

unless it is exactly divisible by 100, unless it is divisible by 400. For example, the year 2004 is a leap year, the year 1900 is not a leap year, and the year 2000 is a leap year.

Compute the number of leap years between the years 1500 and 2010.

WARNING: This is not necessarily the best solution.

```
>> ((2000 - 1500)/4 + 1 + 2) - ((2000 - 1500)/100 + 1) + ((2000-1600)/400+1)
```

» **14.** A very powerful approximation for π was developed by a brilliant mathematician named Srinivasa Ramanujan. The approximation is the following:

$$\frac{1}{\pi} \approx \frac{2\sqrt{2}}{9801} \sum_{k=0}^{N} \frac{(4k)!(1103 + 26390k)}{(k!)^4 396^{4k}}.$$

Use Ramanujan's formula for $N = 0$ and $N = 1$ to approximate π. Be sure to use format long. Compare your approximation with MATLAB's stored value for `pi`. Hint: $0! = 1$ by definition.

```
>> format long
>> (2*sqrt(2)*1103/9801)^-1
>> ((2*sqrt(2)/9801)*(1103 + factorial(4)*(1103 + 26390)/(396^4)))^-1
```

» **15.** The hyperbolic sin or sinh is defined in terms of exponentials as $\sinh(x) = \frac{\exp(x)-\exp(-x)}{2}$.

Compute sinh for $x = 2$ using exponentials. Verify that the result is indeed the hyperbolic sin using MATLAB's built-in function `sinh`.

```
>> 0.5*(exp(2) - exp(-2))
>> sinh(2)
```

» **16.** Verify that $\sin^2(x) + \cos^2(x) = 1$ for $x = \pi, \frac{\pi}{2}, \frac{\pi}{4}, \frac{\pi}{6}$. Use format long.

```
>> format long
>> sin(pi)^2 + cos(pi)^2
>> sin(pi/2)^2 + cos(pi/2)^2
>> sin(pi/4)^2 + cos(pi/4)^2
>> sin(pi/6)^2 + cos(pi/6)^2
```

» **17.** Call the help function for the function `sind`. Use `sind` to compute the $\sin 87°$.

```
>> help sind
>> sind(87)
```

18. Write a MATLAB statement that generates the following error:

"Undefined function or method 'sni' for input arguments of type 'double'."

Hint: sni is a misspelling of the function `sin`.

```
>> sni(pi)
```

19. Write a MATLAB statement that generates the following error:

"Not enough input arguments."

Hint: Input arguments refers to the input of a function (any function), e.g., the input in `sin(pi/2)` is `pi/2`.

```
>> sin()
```

20. Write a MATLAB statement that generates the following error:

"Expression or statement is incorrect–possibly unbalanced (, {, or [."

```
>> (3 + 5
```

21. If P is a logical expression, the law of non-contradiction states that P AND (NOT P) is always false. Verify this for P true and P false.

TRUE AND NOT(TRUE) = TRUE AND FALSE = FALSE.

FALSE AND NOT(FALSE) = FALSE AND TRUE = FALSE.

22. Let P and Q be logical expressions. De Morgan's rule states that NOT (P OR Q) = (NOT P) AND (NOT Q) and NOT (P AND Q) = (NOT P) OR (NOT Q). Generate the truth tables for each statement to show that De Morgan's rule is always true.

For the first row and first column indicating P and Q true, respectively, and the second row and second column indicating P and Q false, respectively, the truth tables for each side of De Morgan's rule will be the following:

$$\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}.$$

23. Under what conditions for P and Q is (P AND Q) OR (P AND (NOT Q)) false?

Whenever P is false.

24. Construct an equivalent logical expression for OR using only AND and NOT.

NOT((NOT P) AND (NOT Q))

25. Construct an equivalent logical expression for AND using only OR and NOT.

NOT ((NOT P) OR (NOT Q))

26. The logical operator XOR has the following truth table:

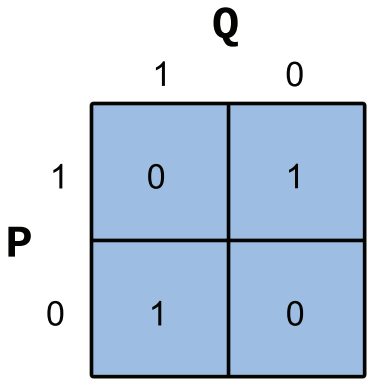

Fig. 1.1. XOR Truth Table.

Construct an equivalent logical expression for XOR using only AND, OR, and NOT that has the same truth table.

(P AND (NOT Q)) OR ((NOT P) AND Q)

27. Do the following calculation at the MATLAB command prompt. Give answers accurate to *16 digits*.

$$e^2 \sin \pi/6 + log_e(3) \cos \pi/9 - 5^3$$

```
>> format long
>> exp(2)*sin(pi/6) + log(3)*cos(pi/9) - 5^3
```

28. Do the following logical and comparison operations at the MATLAB command prompt. You may assume that P and Q are logical expressions.

For $P = 1$ and $Q = 1$: Compute NOT(P) AND NOT(Q).

For $a = 10$ and $b = 25$: Compute $(a < b)$ AND $(a == b)$.

```
>> ~1 & ~1
>> (10 < 25) & (10 == 25)
```