Selected Solutions for Exercises in Numerical Methods with MATLAB: Implementations and Applications

Gerald W. Recktenwald

Chapter 2
Interactive Computing with MATLAB

The following pages contain solutions to selected end-of-chapter Exercises from the book Numerical Methods with MATLAB: Implementations and Applications, by Gerald W. Recktenwald, © 2000, Prentice-Hall, Upper Saddle River, NJ. The solutions are © 2000 Gerald W. Recktenwald. The PDF version of the solutions may be downloaded or stored or printed only for noncommercial, educational use. Repackaging and sale of these solutions in any form, without the written consent of the author, is prohibited.

The latest version of this PDF file, along with other supplemental material for the book, can be found at www.prenhall.com/recktenwald.
2–1 Use the `lookfor` command to search for functions associated with the string “max.” From the list of functions returned, use the `help` facility to determine the function that finds the maximum of all entries in a matrix. Apply this function to find the largest entry in the following matrices:

\[
A = \begin{bmatrix} 1 & -5 & -2 \\ 3 & 4 & -9 \\ -7 & 2 & 6 \end{bmatrix} ; \quad B = \begin{bmatrix} \sin(1) & \sin(-5) & \sin(-2) \\ \sin(3) & \sin(4) & \sin(-9) \\ \sin(-7) & \sin(2) & \sin(6) \end{bmatrix}
\]

**Solution:** Given a vector as input, the `max` function returns the maximum element. Given a matrix as input, the `max` function returns a row vector containing the maximum element in each column of the matrix. To find the maximum element in the matrix, apply `max` twice. Notice that the arguments of the sine function in matrix \(B\) are the elements of \(A\).

```matlab
>> lookfor max % ... returns long list of built-in functions
>> help max % ... returns documentation on max function
>> A = [ 1 -5 -2; 3 4 -9; -7 2 6];
>> max(max(A))
ans =
   6

>> B = sin(A)
B =
 0.8415 0.9589 -0.9093
 0.1411 -0.7568 -0.4121
-0.6570 0.9093 -0.2794
>> max(max(B))
ans =
    0.9589
```

2–5 Use the `linspace` function to create vectors identical to those obtained with the statements that follow. Use multiple statements where necessary. (Use MATLAB’s built-in `norm` function to test whether two vectors are equal without printing the elements.)

(a) \(x = 0:10\)
(b) \(x = 0:0.2:10\)
(c) \(x = -12:12\)
(d) \(x = 10:-1:1\)

**Partial Solution:** The table below gives equivalent `linspace` expressions for the colon notation expressions in the first column. To test whether these statements are correct, enter the commands to create \(x\) and \(y\), and then compute `norm(x-y)`.

<table>
<thead>
<tr>
<th>colon notation</th>
<th><code>linspace</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>(x = 0:10)</td>
<td>(y = linspace(0,10,11))</td>
</tr>
<tr>
<td>(x = -12:12)</td>
<td>(y = linspace(-12,12,25))</td>
</tr>
</tbody>
</table>

Copyright © 2000, Gerald W. Recktenwald. Photocopying is permitted only for non-commercial educational purposes.
2–9 Given the row vector \( x = [10 \ 9 \ 8 \ 7] \) and column vector

\[
y = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}
\]

write at least two different ways to compute the row vector \( z \) defined by \( z_i = x_i - y_i \). Your answers should take only one assignment operation. Do not, for example, explicitly write out equations for all of the elements of \( z \).

**Partial Solution:** One of the less obvious ways to create \( z \) is \( z = (x' - y)' \).

2–15 Use the `eye` and `fliplr` functions to create the matrix

\[
E = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}
\]

Does the same trick work with `flipud`?

**Typographical Error:** `fliplr` should be `fliplr`.

**Solution:** \( E = \text{fliplr}(	ext{eye}(3)) \). Yes, \( E = \text{flipud}(	ext{eye}(3)) \) also works.

2–24 Plot \( \sin \theta \) versus \( \theta \) for 60 points in the interval \( 0 \leq \theta \leq 2\pi \). Connect the points with a dashed line and label the points with open circles. (Hint: Users of MATLAB version 4 will need to plot the data twice in order to combine the symbol and line plots.)

**Partial Solution:** The correct MATLAB statements produce the following plot
2–25 Create a plot of the response of a second-order system with $\zeta = 0$, $\zeta = 0.3$, and $\zeta = 0.9$. Use the formula in Example 2.2, and combine the response curves for all three $\zeta$ values on the same plot. Label the axes and identify the curves with a legend.

**Partial Solution:** The correct MATLAB statements produce the following plot

![Plot of solution to Exercise 2–25.](image-url)
2–30 Data in the table that follows were obtained from an experiment in which the theoretical model is \( y = 5x \exp(-3x) \). The \( x_m \) and \( y_m \) values were measured, and the \( \delta_y \) values were obtained from an uncertainty analysis. Use the built-in `errorbar` function to create a plot of the experimental data with error bars. Use the `hold on` and `plot` functions to overlay a plot of the measured data with the theoretical model. The data are stored in the `xydy.dat` file in the `data` directory of the NMM toolbox.

<table>
<thead>
<tr>
<th>( x_m )</th>
<th>0.010 0.223 0.507 0.740 1.010 1.220 1.530 1.742 2.100</th>
</tr>
</thead>
<tbody>
<tr>
<td>( y_m )</td>
<td>0.102 0.620 0.582 0.409 0.312 0.187 0.122 0.081 0.009</td>
</tr>
<tr>
<td>( \delta_y )</td>
<td>0.0053 0.0490 0.0671 0.0080 0.0383 0.0067 0.0417 0.0687 0.0589</td>
</tr>
</tbody>
</table>

**Partial Solution:** The data can be read with the following statements

\[
\text{D = load('xydy.dat'); x = D(:,1); y = D(:,2); dy = D(:,3);}
\]

The desired plot is shown below.

![Plot of solution to Exercise 2–30.](image-url)