Scalar and Array Operations

Computations in MATLAB typically require wide variety of arithmetic computations between scalars, vectors, and matrices.

Scalar Operations

• Scalar operations are the most obvious if you have programmed in C, Fortran, Basic, etc.

<table>
<thead>
<tr>
<th>Operation</th>
<th>Algebraic Form</th>
<th>MATLAB syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>addition</td>
<td>$a + b$</td>
<td>a + b</td>
</tr>
<tr>
<td>subtraction</td>
<td>$a - b$</td>
<td>a - b</td>
</tr>
<tr>
<td>multiplication</td>
<td>$a \times b$</td>
<td>a*b</td>
</tr>
<tr>
<td>division</td>
<td>$a \div b$</td>
<td>a/b</td>
</tr>
<tr>
<td>exponentiation</td>
<td>$a^b$</td>
<td>a^b</td>
</tr>
</tbody>
</table>

» % Assign constants to a and b:
» a = 3.1; b = 5.234;
» c = a + b
  c = 8.3340
» c = a/b
  c = 0.5923
» c = a^b
  c = 373.0672
» c = c + 1
  c = 374.0672
Array Operations

- When working with vectors and matrices we have the choice of performing operations **element-by-element** or according to the rules of matrix algebra.
- In this section we are only interested in element-by-element operations.
- Basically element-by-element operations on vectors matrices are the same as those of Table 2.1 except must be added before the operator.

<table>
<thead>
<tr>
<th>Operation</th>
<th>Algebraic Form</th>
<th>MATLAB syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>addition</td>
<td>$a + b$</td>
<td>a + b</td>
</tr>
<tr>
<td>subtraction</td>
<td>$a - b$</td>
<td>a - b</td>
</tr>
<tr>
<td>multiplication</td>
<td>$a \times b$</td>
<td>a.*b</td>
</tr>
<tr>
<td>division</td>
<td>$a \div b$</td>
<td>a./b</td>
</tr>
<tr>
<td>exponentiation</td>
<td>$a^b$</td>
<td>a.^b</td>
</tr>
</tbody>
</table>

- Another related case is when a scalar operates on a vector or matrix.
- In this case the scalar is applied to each vector or matrix element in a like fashion.
Examples:

```matlab
» A = [1 3 5 7]
A =
 1  3  5  7
» B = 2*A % Scalar operating on a vector
B =
 2  6 10 14
» B = 2*A
B =
 2  6 10 14
» C = B./A % Vector-to-vector point wise
C =
 2  2  2  2
» D = A.^3
D =
 1 27 125 343
```

Operator Precedence

- The order in which expressions are evaluated in MATLAB is fixed according to Table 2.3

<table>
<thead>
<tr>
<th>Precedence</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>parenthesis, innermost first</td>
</tr>
<tr>
<td>2</td>
<td>exponentiation, left to right</td>
</tr>
<tr>
<td>3</td>
<td>multiplication and division, left to right</td>
</tr>
<tr>
<td>4</td>
<td>addition and subtraction, left to right</td>
</tr>
</tbody>
</table>

- As long as there are matching left and right parenthesis there is no danger in *over doing it*; the parenthesis help insure that operations are done in the proper order

Example: Find the area of the following shape

- There are several ways to solve this problem, but one that
comes to mind is to find the area to the square and then subtract the area of the two right triangles

- Recall that the area of a triangle is $\frac{1}{2} \cdot \text{base} \cdot \text{height}$

$$
\text{Area} = (15 \times 15) - \left[ \frac{1}{2} (15 - 4) \cdot 11 + \frac{1}{2} (15 - 6) \cdot 9 \right]
$$

- Code in MATLAB as follows:

```matlab
Area = (15*15) - 1/2*(15-4)*11+(15-6)*9
Area = 124
```

**Numerical Limitations**

- The usable range of numbers MATLAB can work with is from $10^{-308}$ to $10^{308}$

- If overflow (number too large) occurs MATLAB indicates a result of $\infty$, meaning infinity
• If underflow (number too small) occurs MATLAB indicates a result of 0

Example:

```matlab
» a = 5.00e200; % same as 5*10^200
» b = 3.00e150; % same as 3*10^150
» a*b
    ans =   Inf % should be 15.0e350
» 1/a * 1/b
    ans =     0 % should be (1/5)*(1/3)e-350
```

• Other special cases occur when the conditions of nearly 0 or 0 occur

• For the first case MATLAB gives an error message of divide by zero and returns Inf

• In the second case MATLAB gives an error message of divide by zero and returns NaN

Example:

```matlab
» 1e-30/1e-309 %according to MATLAB like somthing/0
    Warning: Divide by zero.
    ans =    Inf %IEEE notation for 'infinity'
» 1e-309/1e-309 %according to MATLAB like 0/0
    Warning: Divide by zero.
    ans =    NaN %IEEE notation for 'not a number'
```