Lab 2: Arithmetic Operations on Arrays

Element-by-Element Operations

- They are done between a scalar and an array, or between corresponding elements of arrays.

Scalar-Array Operations

```
>> A = [2 9; 5 -7]

A =

     2     9
     5    -7

>> A+3

ans =

     5    12
     8    -4

>> A-3

ans =

    -1     6
     2   -10

>> A*3

ans =

     6    27
    15   -21

>> A/3

ans =

    0.6667    3.0000
    1.6667   -2.3333
```

Array-Array Operations

- The mathematical operation is repeated between corresponding elements in arrays.
- Arrays should be or equal dimensions.

```
>> A = [6 -2;10 3];
>> B = [9 8; -12 14];
>> A+B

ans =

    15     6
    -2    17
```

- For multiplication or division use .* and ./

```
>> x = [2 4 -5];
>> y = [-7 3 -8];
>> z = x.*y

z =

   -14    12    40
```

- The built-in Matlab functions automatically operate on arrays in an element-by-element fashion.

```
>> x = [4 16 25]

x =

     4    16    25

>> sqrt(x)

ans =

     2     4     5
```

- However, when multiplying or dividing these functions, or when raising them to a power, you must use element-by-element operations. For example to compute: $z = (e^y \sin x)\cos^2 x$ you must type:
  `z = exp(y).*sin(x).*(cos(x)).^2`
  You will get an error if the size of x and y is not the same. The result z will have the same size as x and y.
- The power function (.^)

```
>> x = [3 5 8];
>> x.^3

ans =

    27   125   512


>> p = [2 4 5];
>> 3.^p

ans =

     9    81   243
```

Example:

Evaluate $f(x,y) = xy^2+8x$ at (1,2) (3,1) (5,9)

```
>> x = [1 3 5];
>> y = [2 1 9];
>> f = x.*(y.^2)+8*x

f =

    12    27   445
```

Matrix Multiplication

Here, the number of columns in the first matrix should equal the number of rows in the second matrix. The resulting matrix will have the same number of rows as the first matrix and the same number of columns as the second matrix.

```
>> a = [6 -2; 10 3; 4 7];
>> b = [9 8; -5 12];
>> a*b

ans =

    64    24
    75   116
     1   116
```

Matrix multiplication is not commutative. AB ≠ BA.

```
>> A = [6 -2; 10 3];
>> B = [9 8; -12 14];
>> A*B

ans =

    78    20
    54   122

>> B*A

ans =

   134     6
    68    66
```

Matrix Division

- Matrix Division is a more challenging topic than matrix multiplication.
- Matrix Division uses both right and left division operators / and \ for various applications.
- One application is solving linear algebraic equations:
  $6x+12y+4z = 70$
  $7x-2y+3z = 5$
  $2x+8y-9z = 64$

  ```
  >> A = [6 12 4;7 -2 3;2 8 -9];
  >> B = [70;5;64];
  >> A\B

  ans =

       3
       5
      -2
  ```

  Another Method:

  ```
  >> A = [6 12 4;7 -2 3;2 8 -9];
  >> B = [70;5;64];
  >> inv(A)*B

  ans =

      3.0000
      5.0000
     -2.0000
  ```

Matrix Exponentiation

- A^n is equivalent to multiplying the matrix by itself n times.
- A should be a square matrix.
- A^B where B is a matrix is not defined.

Special Matrices

- Identity matrix: eye(n) eye(m,n)
- All-ones matrix: ones(n) ones(m,n)
- All-zeros matrix: zeros(n) zeros(m,n)

Polynomial Operations using Arrays

- The polynomial:

$$f(x) = a_1x^n + a_2x^{n-1} + \cdots + a_nx + a_{n+1}$$

  Can be represented in Matlab by:
  ```
  >> A = [a1,a2,...,an+1]
  ```

- Use the function roots(A) to find the polynomial roots:

```
>> A = [1 5 6];
>> c = roots(A)

c =

   -3.0000
   -2.0000
```

- Use the function poly(c) to find the coefficients of the polynomial from its roots:

```
>> c = [-1 2];
>> poly(c)

ans =

     1    -1    -2
```

This means the answer is x^2-x-2

- Multiplication of Polynomials: Use the function conv(a,b) to compute the product of the two polynomials described by the coefficient arrays a and b. The two polynomials need not be the same degree. The result is the coefficient array of the product polynomial.

- Division of Polynomials: Use the function [q,r] = deconv(num,den) to compute the result of dividing a numerator polynomial, whose coefficient array is num, by a denominator polynomial represented by the coefficient array den. The quotient polynomial is given by the coefficient array q, and the remainder polynomial is given by the coefficient array r.

Example:

```
>> a = [9 -5 3 7];
>> b = [6 0 2];
>> product = conv(a,b)

product =

    54   -30    36    32     6    14

>> [Q,R] = deconv(a,b)

Q =

    1.5000   -0.8333


R =

         0          0          0    8.6667
```

- Evaluation of Polynomials: The function polyval(a,x) evaluates a polynomial at specified values of its independent variable x, which can be a matrix or a vector. The polynomial's coefficients of descending powers are stored in the array a. The result is the same size as x.

```
>> a = [9 -5 3 7];
>> x = [-2 3 4 5];
>> f = polyval(a,x)

f =

   -91   214   515   1022
```