Computer Programming with MATLAB

VANDERBILT UNIVERSITY

Lesson 2: Matrices and Operators

by

Akos Ledeczi and Mike Fitzpatrick

Introduction to Arrays and Matrices

- Array
  - Any set of numbers arranged in a rectangular pattern.

Example—

A page with six rows of four numbers each is a two-dimensional array

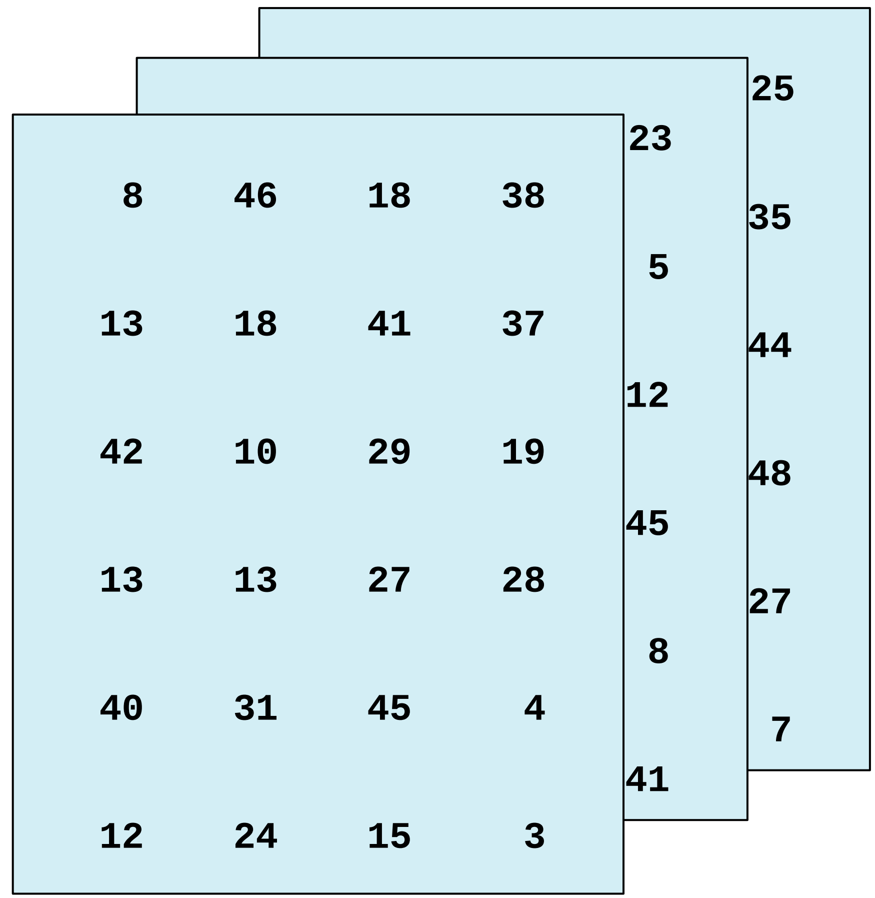
10	14	48	25
24	34	17	35
22	33	29	44
32	8	11	48
35	6	37	27
37	25	13	7

Introduction to Arrays and Matrices

- Array
  - Any set of numbers arranged in a rectangular pattern.

Three-dimensional Example—

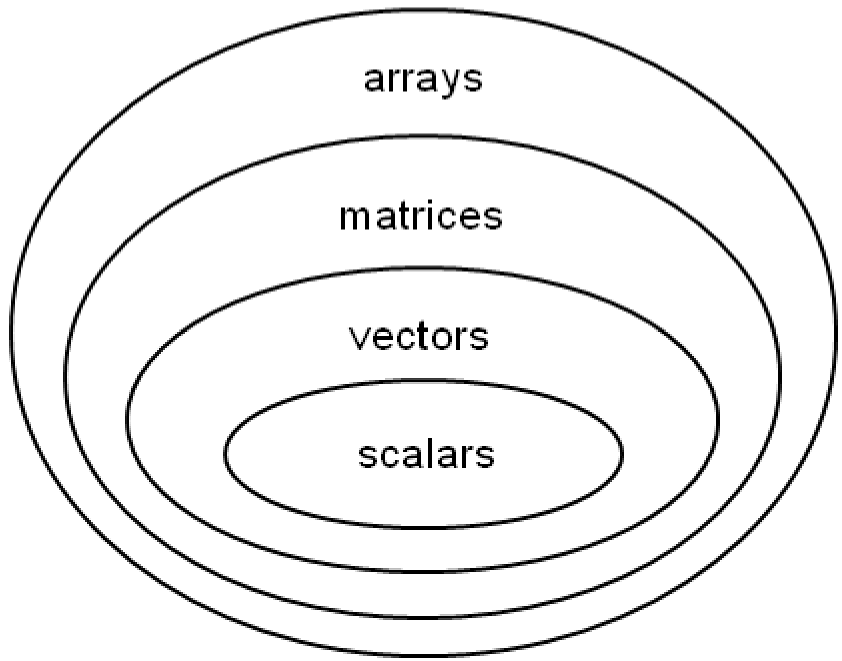
A stack of such pages



Introduction to Arrays and Matrices

- Higher dimensions are uncommon
- The most common have special names:
  - 2D array = "matrix" (plural is "matrices")
  - 1D array = "vector"
- Most ingenious part of Cleve Moler's invention of MATLAB was the way he set it up to deal with matrices.
- MATLAB stands for "Matrix Laboratory"!

Arrays and Matrices



Rows and Columns

```
>> X = [1:4; 5:8; 9:12];
```

$$\begin{array}{r}1:\\2:\\3:\end{array}\begin{bmatrix}1 & 2 & 3 & 4\\5 & 6 & 7 & 8\\9 & 10 & 11 & 12\end{bmatrix}\qquad \text{rows}$$

VANDERBILT UNIVERSITY

Rows and Columns

VANDERBILT UNIVERSITY

```
>> X = [1:4; 5:8; 9:12];
```

$$\begin{array}{c} \begin{array}{cccc} 1: & 2: & 3: & 4: \end{array} \\ \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{bmatrix} \end{array}$$

columns

Indexing

VANDERBILT UNIVERSITY

```
>> X = [1:4; 5:8; 9:12];
>> X(2,3)
```

$$\begin{array}{cc} & \begin{array}{cccc} & & 3: & \end{array} \\ 2: & \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{bmatrix} \end{array}$$

```
>> ans =

         7
```

Array Addition

- Z = X + Y means
  - Z(m,n) = X(m,n) + Y(m,n) for all valid m and n

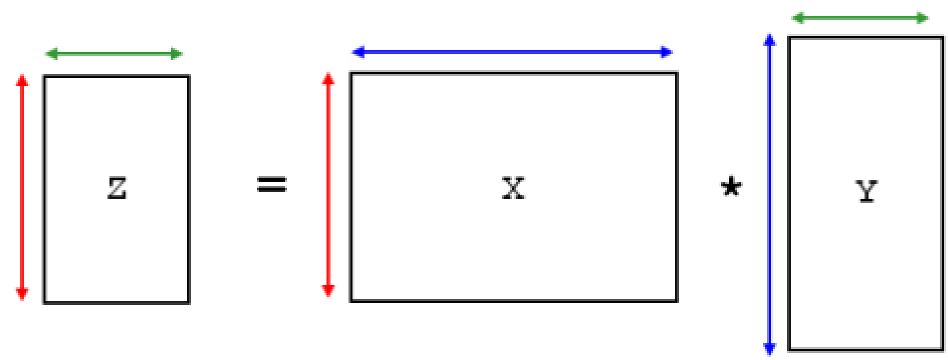
```
Z( 1 , 1 )   =   X( 1 , 1 )   +   Y( 1 , 1 )
Z( 1 , 2 )   =   X( 1 , 2 )   +   Y( 1 , 2 )    1st row
. . .
Z( 1 ,end)   =   X( 1 ,end)   +   Y( 1 ,end)

Z( 2 , 1 )   =   X( 2 , 1 )   +   Y( 2 , 1 )
Z( 2 , 2 )   =   X( 2 , 2 )   +   Y( 2 , 2 )    2nd row
. . .
Z( 2 ,end)   =   X( 2 ,end)   +   Y( 2 ,end)

. . .

Z(end, 1 )   =   X(end, 1 )   +   Y(end, 1 )
Z(end, 2 )   =   X(end, 2 )   +   Y(end, 2 )    lastrow
. . .
Z(end,end)   =   X(end,end)   +   Y(end,end)
```

Matrix Multiplication

- Different from Array Multiplication!
- **`Z = X * Y`** means that for all valid m and n

$$\texttt{Z(m,n)} = \sum_{k} \texttt{X(m,k)Y(k,n)}$$

- Not always legal:
  - Inner dimensions of X and Y must be the same



Array Division

- `Z = X./Y`
  - means that for each m and n, Z(m,n) = X(m,n)/Y(m,n)
- `Z = X.\Y`
  - means that for each m and n, Z(m,n) = Y(m,n)/X(m,n)
- Try these out in MATLAB on your own!
- Matrix division is a complicated concept in linear algebra, so we are not covering it here
  - But you can check out the advanced concepts of the textbook for detailed explanation

Precedence

- **x = a + b + c**
  - order does not matter with addition
- **y = c + a * b** is not the same as
- **y = (c + a) * b**
- Multiplication has priority over addition
  - In programming, this is called *precedence*

PRECEDENCE	OPERATOR
0	Parentheses: (...)
1	Exponentiation ^ and Transpose '
2	Unary +, Unary −, and logical negation: ~
3	Multiplication and Division (array and matrix)
4	Addition and Subtraction
5	Colon operator :

Precedence Table

Associativity

VANDERBILT UNIVERSITY

- **x = a + b + c**
- **x = a * b * c**
  - order does not matter with addition or multiplication
- **y = a ^(b ^ c)** is not the same as
- **y = (a ^ b)^ c**
- In programming, the order in which operators of the same precedence are executed is called *associativity*
- In MATLAB, it is left to right
- **y = a ^ b ^ c** is the same as
- **y = (a ^ b) ^ c**