Previous Lecture:
- Discrete vs. continuous; finite vs. infinite
- Vectorized operations

Today’s Lecture:
- 2-d array—matrix

Announcements:
- Discussion this week in the classrooms as listed in the roster
- Prelim 1 tonight at 7:30pm
  - Last names A-G: Ives Hall 105
  - Last names H-K: Ives Hall 217
  - Last names L-Z: Ives Hall 305

Initialize arrays if dimensions are known
... instead of “building” the array one component at a time

```matlab
% Initialize y
x=linspace(a,b,n);
y=zeros(1,n);
for k=1:n
    y(k)=myF(x(k));
end
```

```matlab
% Build y on the fly
x=linspace(a,b,n);
y(k)=myF(x(k));
end
```

Much faster for large n!

Vectorized code
--- a Matlab-specific feature

- Code that performs element-by-element arithmetic/relation/logical operations on array operands in one step

- Scalar operation: $x + y$
  - where $x, y$ are scalar variables

- Vectorized code: $x + y$
  - where $x$ and/or $y$ are vectors. If $x$ and $y$ are both vectors, they must be of the same shape and length

Vectorized addition

```
\[
x \begin{bmatrix} 2 & 1.5 & 8 \end{bmatrix} \\
+ \begin{bmatrix} 1 & 2 & 0 & 1 \end{bmatrix}
\]
```

```
\[
= \begin{bmatrix} 3 & 3.5 & 9 \end{bmatrix}
```

Matlab code: $z = x + y$

Vectorized multiplication

```
\[
a \begin{bmatrix} 2 & 1.5 & 8 \end{bmatrix} \\
\times \begin{bmatrix} 1 & 2 & 0 & 1 \end{bmatrix}
\]
```

```
\[
= \begin{bmatrix} 2 & 2 & 0 & 8 \end{bmatrix}
```

Matlab code: $c = a \times b$

Vectorized element-by-element arithmetic operations on arrays

See full list of ops in §4.1

A dot (.) is necessary in front of these math operators
### Shift

\[
\begin{array}{c}
\times 3 \\
+ 2 1 .5 8 \\
= 5 4 3.5 11 \\
\end{array}
\]

Matlab code: \( z = x + y \)

### Reciprocate

\[
\begin{array}{c}
\times 1 \\
/ 2 1 .5 8 \\
= .5 1 2 .125 \\
\end{array}
\]

Matlab code: \( z = x ./ y \)

### Vectorized

Element-by-element arithmetic operations between an array and a scalar

\[
\begin{array}{c}
+ \\
- \\
\times \\
/ \\
\end{array}
\]

A dot (.) is necessary in front of these math operators

The dot in \( + \), \( \times \), \( \div \) not necessary but OK

### Can we plot this?

\[
f(x) = \frac{\sin(5x) \exp(-x/2)}{1 + x^2} \quad \text{for} \ -2 \leq x \leq 3
\]

Yes!

\[
x = \text{linspace}(-2,3,200); \\
y = \sin(5*x) .* \exp(-x/2) ./ (1 + x.^2); \\
\text{plot}(x,y)
\]

### Element-by-element arithmetic operations on arrays...

Also called “vectorized code”

\[
x = \text{linspace}(-2,3,200); \quad \text{\( x \) and \( y \) are vectors} \\
y = \sin(5*x) .* \exp(-x/2) ./ (1 + x.^2); \quad \text{Contrast with scalar operations that we’ve used previously…}
\]

\[
a = 2.1; \quad \text{\( a \) and \( b \) are scalars} \\
b = \sin(5*a); \\
The operators are (mostly) the same; the operands may be scalars or vectors. \\
When an operand is a vector, you have “vectorized code.”
\]

### Storing and using data in tables

A company has 3 factories that make 5 products with these costs:

\[
\begin{array}{cccccc}
C & 10 & 36 & 22 & 15 & 62 \\
12 & 35 & 20 & 12 & 66 \\
13 & 37 & 21 & 16 & 59 \\
\end{array}
\]

What is the best way to fill a given purchase order?
2-d array: matrix

- An array is a named collection of like data organized into rows and columns
- A 2-d array is a table, called a matrix
- Two indices identify the position of a value in a matrix, e.g., mat(r,c)
  refers to component in row r, column c of matrix mat
- Array index starts at 1
- Rectangular: all rows have the same # of columns

Creating a matrix

- Built-in functions: ones, zeros, rand
  - E.g., zeros(2,3) gives a 2-by-3 matrix of 0s
  - E.g., zeros(2) gives a 2-by-2 matrix of 0s
- “Build” a matrix using square brackets, [], but the dimension must match up:
  - [x y] puts y to the right of x
  - [x; y] puts y below x
  - [4 0 3; 5 1 9] creates the matrix
  - [4 0 3; ones(1,3)] gives
  - [4 0 3; ones(3,1)] doesn’t work

Working with a matrix:

- size and individual components

  Given a matrix M
  \[
  \begin{array}{cccc}
  2 & -1 & 5 & 0 \\
  3 & 8 & 6 & 7 \\
  5 & -3 & 8.5 & 9 \\
  52 & 81 & .5 & 7 \\
  \end{array}
  \]

  \[
  [nr, nc]= size(M) \quad \% \; nr \; is \; # \; of \; rows,
  \% \; nc \; is \; # \; of \; columns
  \]

  nr = size(M, 1) \quad \% \; # \; of \; rows

  nc = size(M, 2) \quad \% \; # \; of \; columns

  M(2,4) = 1;
  disp(M(3,1))
  M(1,nc) = 4;

Example: minimum value in a matrix

function val = minInMatrix(M)
  \%
  \% val is the smallest value in matrix M

Pattern for traversing a matrix M

% Given an nr-by-nc matrix M.
% What is A?
for r = 1: nr
  for c = 1 : nc
    A(c,r) = M(r,c); 
  end
end

A  A is M with the columns in reverse order
B  A is M with the rows in reverse order
C  A is the transpose of M
D  A and M are the same
% Given an nr-by-nc matrix M.
% What is A?
for r= 1: nr
    for c= 1: nc
        A(c,r)= M(r,c);
    end
end

function A = RandomLinks(n)
% A is n-by-n matrix of 1s and 0s
% representing n webpages
A = zeros(n,n);
for i=1:n
    for j=1:n
        r = rand(1);
        if i~=j && r<= 1/(1 + abs(i-j));
            A(i,j) = 1;
        end
    end
end

Matrix example: Random Web

- N web pages can be represented by an N-by-N Link Array A.
- A(i,j) is 1 if there is a link on webpage j to webpage i.
- Generate a random link array and display the connectivity:
  - There is no link from a page to itself
  - If i≠j then A(i,j) = 1 with probability \( \frac{1}{1+|i-j|} \)
  - There is more likely to be a link if i is close to j

Represent the web pages graphically...

100 Web pages arranged in a circle.
Next display the links....
% Given an n-by-m matrix A.
% What is this operation?
for g = 1:n
    for h = 1:floor(m/2)
        A(g,h) = A(g, m-h+1);
    end
end

A. Reflect the right half of A onto the left half
B. Reflect the bottom half of A onto the top half