Topics: Two-dimensional array—matrix, file input/output (I/O)

Reading (ML): Sec 2.3, 2.7, review Sec 2.2, 2.4

Matrix creation, access, manipulation

\[
m = [1 \ 2 \ 3 \ 4; \ 5 \ 6 \ 7 \ 8] \ % \ 2\text{-by-}4 \ matrix
\]
\[
[nr, nc] = \text{size}(m)
\]
\[
m = [m; \ \text{zeros}(1, nc)]
\]
\[
m = [m \ m]
\]
\[
v = 1:6
\]
\[
\text{newm} = [m \ v']
\]
\[
\text{newm} = \text{newm}'
\]
\[
m1 = \text{rand}(4, 3) \ % \ 4\text{-by-}3 \ random \ matrix \ (uniform \ dist.)
\]
\[
tmp = m1(3, 2) \ % \ cell \ in \ 3rd \ row, \ 2nd \ column
\]
\[
tmp = m1(3:4, :)
\]
\[
tmp = m1(:, 2)
\]
\[
tmp = m1([1 \ 4], :)
\]
\[
tmp = m1(:, [1 \ 3])
\]
\[
tmp = m1([1 \ 4], [1 \ 3])
\]
load \text{matrix.dat} \ % \ an \ ASCII \ data \ file \ called \ \text{matrix.dat}
load \text{mmatrix} \ % \ a \ MATLAB \ data \ file \ called \ \text{mmatrix.mat}
help save

Processing data in a matrix

Write a function \text{mysum} that takes as an input argument a matrix \text{m} and returns the sum all the values in \text{m}. Do not use any MATLAB predefined functions (other than \text{size}) or vectorized code.

\[
\% \text{mysum(m): calculate sum of all entries in matrix m}
\% total: sum of entries in matrix m
\]
\[
[nr, nc] = \text{size}(m);
\]
\[
\% \ initialize \ variable \ for \ accumulating \ the \ sum
\]
\[
\% \ sum \ one \ element \ at \ a \ time
Pattern for traversing a matrix

```matlab
for r = 1:nr
    for c = 1:nc
        % do something
    end
end
```

Note: Need to first assign values to nr (no. of rows), nc (no. of columns)

Example

Calculate the sum of the diagonal elements in a given square matrix `sm`.

```matlab
for r = 1:nr
    for c = 1:nc
        % sum only the diagonal elements
    end
end
```

MATLAB predefined functions

Below are some useful functions that work on matrices. These functions “operate” on each column of the matrix and return a row vector.

`sum, mean, max, min`