L21. More on 2D Arrays

And their connections to
Cell arrays
Structure arrays
Character arrays
Application: Digital Displays

1649
7-by-5 “Dot Matrices”
A “Bit Map” For Each Digit

A “light” is either on or off.

A 7-by-5 matrix of zeros and ones can “tell the whole story”.
Look at Computations with These Bitmaps

First order of business:
Store the 10 bitmaps
Design Decisions

How do we package a particular digit?

numerical array or character array

How do we package the collection of digits?

cell array or structure array

We look at the 4 possibilities.
Storing a Single Bitmap
Can Use a Numerical Array For Each Digit

\[
\begin{bmatrix}
0 & 1 & 1 & 1 & 0; & \ldots \\
1 & 0 & 0 & 0 & 1; & \ldots \\
0 & 0 & 0 & 1 & 0; & \ldots \\
0 & 0 & 1 & 0 & 0; & \ldots \\
0 & 1 & 0 & 0 & 0; & \ldots \\
1 & 0 & 0 & 0 & 0; & \ldots \\
1 & 1 & 1 & 1 & 1 ;
\end{bmatrix}
\]
Can Use a Character Array For Each Digit

\[
A = [ '01110', '10001', '00010', '00100', '01000', '10000', '11111' ];
\]
Storing the 10 Bitmaps in a Cell Array
Can Use a Cell Array this way

\[
M = \begin{bmatrix}
0 & 1 & 1 & 1 & 1 & 0 \\
1 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 \\
\end{bmatrix};
\]

\[D\{2\} = M;\]

Here a cell is a numerical matrix.
With $D\{1\}, \ldots, D\{10\}$

Set Up:

\[
M = D\{k\};
\]
\[
\text{if } M(4,3) == 1
\]
\[
\quad \text{disp('Middle Light is On')}
\]
\[
\text{end}
\]

Here $k$ is initialized and satisfies $1 \leq k \leq 10$
M = [ '01110';...
    '10001';...
    '00010';...
    '00100';...
    '01000';...
    '10000';...
    '11111' ];

D{2} = M;

Here a cell is a char array
With $D{1},...,D{10}$

Set Up:

$M = D{k}$;

if $\text{strcmp}(M(4,3),'1')$
    disp('Middle Light is On')
end

Here $k$ is initialized and satisfies $1 \leq k \leq 10$
Storing the 10 Bitmaps in a Structure Array
Can Use a Struct Array Like This

\[
M = \begin{bmatrix}
0 & 1 & 1 & 1 & 0 \\
1 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 \\
\end{bmatrix};
\]

\[
D(2) = \text{struct}('mat', M);
\]

Here the sole field is a matrix
With $D(1),\ldots,D(10)$

Set Up:

\[
M = D(k).\text{mat};
\]

\[
\text{if } M(4,3) == 1
\]\[
\quad \text{disp(} \text{`Middle Light is On'} \text{)}
\]

\[
\text{end}
\]

Here $k$ is initialized and satisfies $1\leq k\leq 10$
M = [ '01110'; '10001'; '00010'; '00100'; '01000'; '10000'; '11111' ];

D(2) = struct('mat',M);

Here the sole field is a char array
With D(1),...,D(10)
Set Up:

M = D(k).mat
if strcmp(M(4,3),’1’)
    disp(‘Middle Light is On’)  
end

Here k is initialized and satisfies 1<=k<=10
Choice for Storing the Bit Maps

Cell array better than struct array
   No point in having a structure with one field.
Numerical array better than char array
   Plan on doing numerical computations with the bit arrays. Char arrays not handy
function D = TheDigits
D = cell(10,1);
D{1} = [0 0 1 0 0;...
     0 1 1 0 0;...
     0 0 1 0 0;...
     0 0 1 0 0;...
     0 0 1 0 0;...
     0 0 1 0 0;...
     0 0 1 0 0;...
     0 1 1 1 0];

etc
Problem

Produce a cell array of "reverse" digits
Reversing Column Order

Suppose $A$ has 5 columns. If

\[
\begin{align*}
B(:,1) &= A(:,5) \\
B(:,2) &= A(:,4) \\
B(:,3) &= A(:,3) \\
B(:,4) &= A(:,2) \\
B(:,5) &= A(:,1)
\end{align*}
\]

then $B$ is $A$ with its cols reversed.

\[
B(:,k) = A(:,6-k)
\]
A Function to Do the Job

function B = ReverseCol(A)
[p,q] = size(A);
B = zeros(p,q);
for k=1:q
    B(:,k) = A(:,q-k+1);
end
A Cell Array of Reversed Digits

\[
D = \text{TheDigits};
\]

\[
\text{revD} = \text{cell}(10,1)
\]

\[
\text{for } k=1:10
\]

\[
\text{M} = D\{k\};
\]

\[
\text{revM} = \text{ReverseCol}(M);
\]

\[
\text{revD}\{k\} = \text{revM};
\]

\[
\text{end}
\]
The Difference Between Two Bit Maps

\[ C(i,j) = \text{abs}( A(i,j) - B(i,j) ) \]
function C = Difference(A,B)
% A and B are p-by-q arrays.
% C is a p-by-q array with
% C(i,j) = abs(A(i,j)-B(i,j))

[p,q] = size(A);
C = zeros(p,q);
for i=1:p
    for j=1:q
        C(i,j) = abs(A(i,j) - B(i,j));
    end
end
Problem

100000 random digits are displayed in succession on a 7-by-5

How often does each of the 35 “bulbs” go on and off?
Digression: 2D Array Ops

```matlab
>> A = [1 2; 3 4];
>> B = [10 20; 30 40];
>> C = A + B
  C =
      11   22
      33   44
```
Adding Up The Changes

D = TheDigits;
Count = zeros(7,5);
n = 10000;
for k=1:n
    i1 = ceil(10*rand);
    i2 = ceil(10*rand);
    M = Difference(D{i1},D{i2});
    Count = Count + M;
end
## Results

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>41979</td>
<td>31670</td>
<td>17754</td>
<td>31670</td>
<td>41979</td>
<td></td>
</tr>
<tr>
<td>31933</td>
<td>17936</td>
<td>17936</td>
<td>0</td>
<td>41913</td>
<td></td>
</tr>
<tr>
<td>48081</td>
<td>0</td>
<td>17936</td>
<td>17991</td>
<td>47770</td>
<td></td>
</tr>
<tr>
<td>48032</td>
<td>50078</td>
<td>31970</td>
<td>41836</td>
<td>41786</td>
<td></td>
</tr>
<tr>
<td>41818</td>
<td>0</td>
<td>41986</td>
<td>0</td>
<td>41986</td>
<td></td>
</tr>
<tr>
<td>49871</td>
<td>18011</td>
<td>31933</td>
<td>0</td>
<td>41986</td>
<td></td>
</tr>
<tr>
<td>18011</td>
<td>31707</td>
<td>17754</td>
<td>31707</td>
<td>31841</td>
<td></td>
</tr>
</tbody>
</table>