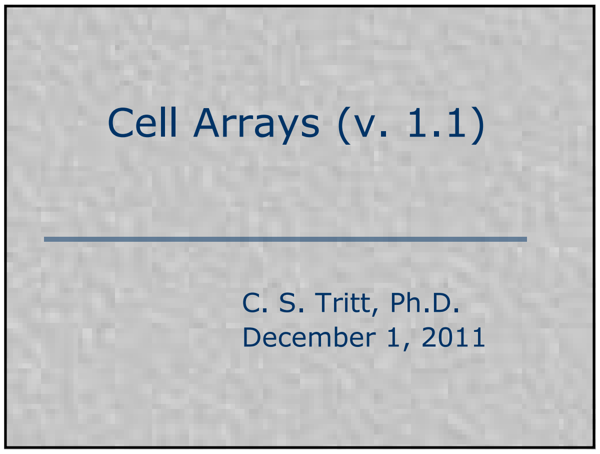
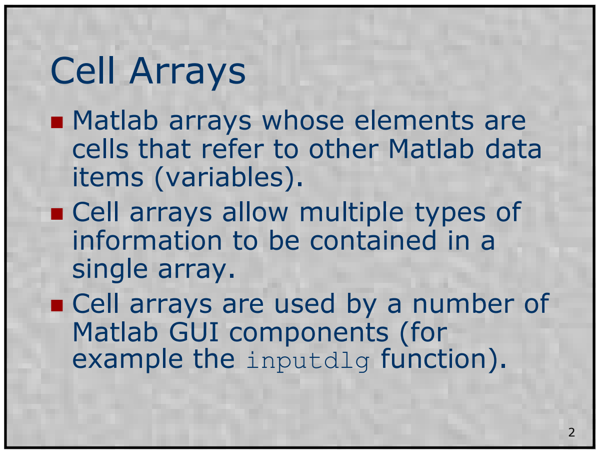


Cell Arrays (v. 1.1)

C. S. Tritt, Ph.D.
December 1, 2011

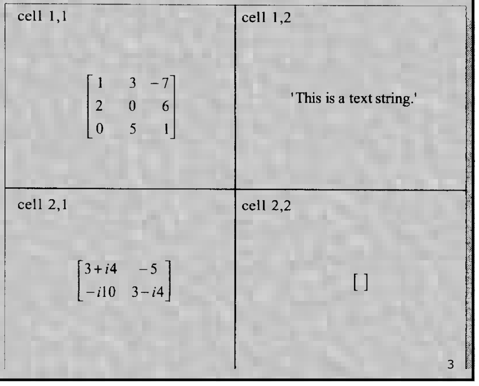


Cell Arrays

- Matlab arrays whose elements are cells that refer to other Matlab data items (variables).
- Cell arrays allow multiple types of information to be contained in a single array.
- Cell arrays are used by a number of Matlab GUI components (for example the `inputdlg` function).

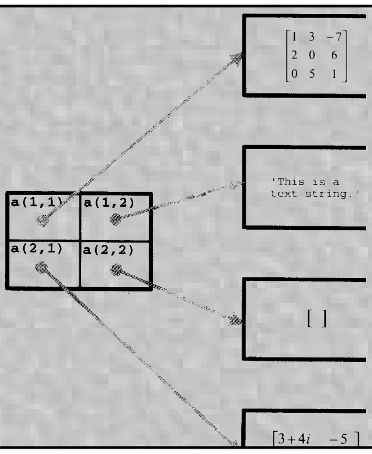
2

Cell Array Schematic


cell 1,1
$\begin{bmatrix} 1 & 3 & -7 \\ 2 & 0 & 6 \\ 0 & 5 & 1 \end{bmatrix}$
cell 1,2
'This is a text string.'
cell 2,1
$\begin{bmatrix} 3+i4 & -5 \\ -i10 & 3-i4 \end{bmatrix}$
cell 2,2
[ ]

3

Cell Arrays Hold References



a(1,1)
a(1,2)
a(2,1)
a(2,2)
$\begin{bmatrix}1 & 3 & -7\\2 & 0 & 6\\0 & 5 & 1\end{bmatrix}$
'This is a text string.'
[ ]
[3+4i -5]

4

Dealing with References

- With references, there is the issue of distinguishing if you are referring to the reference or what is being referred to.
- This results in the need for two approaches for assigning and recalling pointer values.
- This is similar to dealing with pointers in other languages. In some language (like Java) this is automatic. In others (like C and Matlab) it's manual. In C++, it's a little of both!

5

Assigning to Cell Arrays

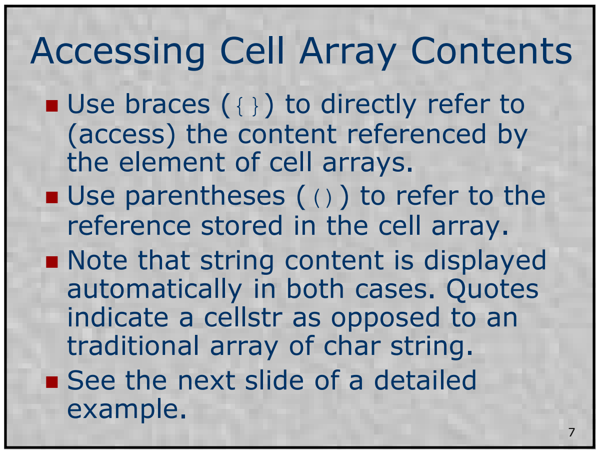
- The following sets of assignments have exactly the same effect:
  - Content indexing:

```
a{1,1} = [ 1 2 3; 4 5 6 ];
a{1,2} = 'Hello World';
```

  - Cell indexing:

```
a(1,1) = {[1 2 3; 4 5 6 ]};
a(1,2) = {'Hello World'};
```

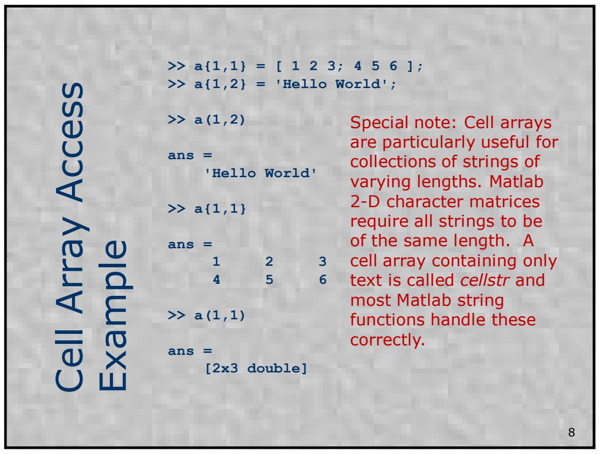
6



Accessing Cell Array Contents

- Use braces ({ }) to directly refer to (access) the content referenced by the element of cell arrays.
- Use parentheses (( )) to refer to the reference stored in the cell array.
- Note that string content is displayed automatically in both cases. Quotes indicate a cellstr as opposed to an traditional array of char string.
- See the next slide of a detailed example.

7

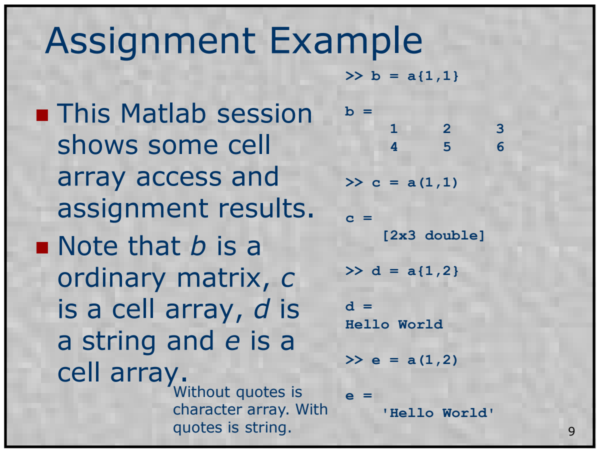


Cell Array Access Example

```
>> a{1,1} = [ 1 2 3; 4 5 6 ];
>> a{1,2} = 'Hello World';

>> a(1,2)

ans =
    'Hello World'

>> a{1,1}

ans =
     1     2     3
     4     5     6

>> a(1,1)

ans =
    [2x3 double]
```

Special note: Cell arrays are particularly useful for collections of strings of varying lengths. Matlab 2-D character matrices require all strings to be of the same length. A cell array containing only text is called *cellstr* and most Matlab string functions handle these correctly.

8



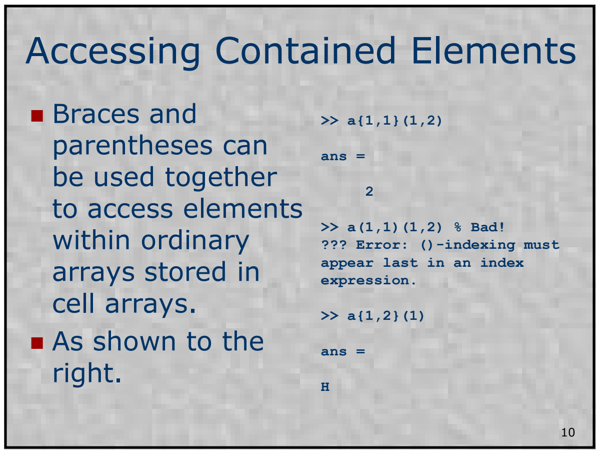
Assignment Example

- This Matlab session shows some cell array access and assignment results.
- Note that *b* is a ordinary matrix, *c* is a cell array, *d* is a string and *e* is a cell array.

Without quotes is character array. With quotes is string.

```
>> b = a{1,1}

b =
     1     2     3
     4     5     6

>> c = a(1,1)

c =
    [2x3 double]

>> d = a{1,2}

d =
Hello World

>> e = a(1,2)

e =
    'Hello World'
```

9

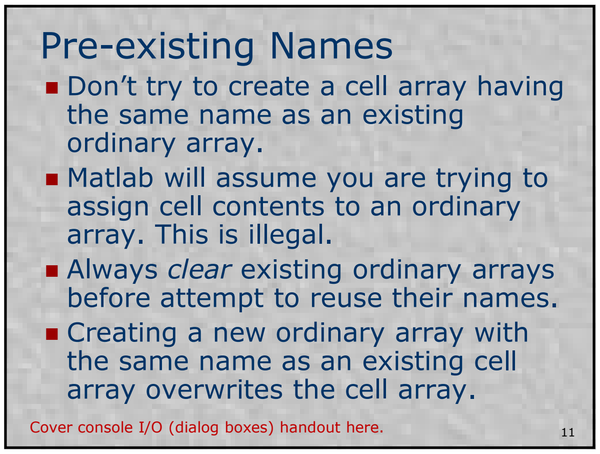


Accessing Contained Elements

- Braces and parentheses can be used together to access elements within ordinary arrays stored in cell arrays.
- As shown to the right.

```
>> a{1,1}(1,2)

ans =

     2

>> a(1,1)(1,2) % Bad!
??? Error: ()-indexing must
appear last in an index
expression.

>> a{1,2}(1)

ans =

H
```

10



Pre-existing Names

- Don’t try to create a cell array having the same name as an existing ordinary array.
- Matlab will assume you are trying to assign cell contents to an ordinary array. This is illegal.
- Always *clear* existing ordinary arrays before attempt to reuse their names.
- Creating a new ordinary array with the same name as an existing cell array overwrites the cell array.

Cover console I/O (dialog boxes) handout here.

11