Data Structures

Chapter 8
**Data Structure:** A variable that stores more than one value

- Matrices/vectors and character arrays are types of data structures.

MATLAB also provides two other types of data structures:

- **Cell Array**
  - Stores values of different types.
  - Values can be strings, chars, doubles, matrices, etc...
  - Numerically Indexed (so you can loop through them)

- **Structure**
  - Stores values of different types
  - Values can be strings, chars, doubles, matrices, etc...
  - Values are stored in “**fields**” (i.e. you can’t loop through them)
  - Useful for making databases
Up until now we have mainly focused on what matrices and character arrays can do

• Some General Rules...
  • Must have consistent #cols
  • Can’t have empty entries
  • Can’t mix variables of different type
  • i.e. you can’t mix doubles, ints, and chars
Common Earth Science Data Sets

**Advanced National Seismic System (ANSS)**
- [http://www.ncedc.org/anss/](http://www.ncedc.org/anss/)
- Near real-time data for all global earthquake events!
- *Not just numbers*

**USGS Current Water Data**
- Near real-time data for streams in all 50 states!
- *Not just numbers*
# Cell Arrays: The Basics

What if you wanted to make a data structure that could hold anything of any size? (assuming you have sufficient RAM)

<table>
<thead>
<tr>
<th>Matrices</th>
<th>Cell Arrays</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Must have consistent dimensions</td>
<td>• An array of cells</td>
</tr>
<tr>
<td>• Must be all numeric (no chars/strings)</td>
<td>• Each cell can hold anything</td>
</tr>
<tr>
<td></td>
<td>• char, double, int, matrix, vector, string, logical</td>
</tr>
<tr>
<td></td>
<td>• Each entry can be any size</td>
</tr>
<tr>
<td></td>
<td>• Unlike matrices, a given cell can be empty</td>
</tr>
<tr>
<td></td>
<td>• Technically, the cell doesn’t hold the value, it just contains a pointer to where the contents are stored.</td>
</tr>
<tr>
<td></td>
<td>• This usually isn’t important for us as Earth scientists.</td>
</tr>
<tr>
<td>Character Arrays (Strings)</td>
<td></td>
</tr>
<tr>
<td>• Must have consistent dimensions (of characters)</td>
<td></td>
</tr>
<tr>
<td>• Must contain only characters</td>
<td></td>
</tr>
</tbody>
</table>
How do I create a cell array?

- Cell arrays are created using curly braces `{ }`.
- Follow the same indexing rules as matrices.
  - If you use `()`, this is called cell indexing.
  - Returns the cell (not the contents) of the specified index.

```matlab
>> myCell = {'Scott', 42, 'fun', 1:4, zeros(3,3)}
```
How do I create a cell array?

- Cell arrays are created using curly braces { }

To get a cell’s contents use curly braces { }

- Called content indexing
- Returns the cell’s contents of the specified index
- Typically, a string, some numeric value, or matrix
Cell Arrays: Pre-Allocating

How do I pre-allocate a cell array?

- Use ‘cell(m,n)’
  - Makes an empty m x n cell array
  - You can later add in the cell contents
  - Essentially the same as pre-allocating a matrix, except cells can be empty

Why should I do this?

- Cell arrays suffer from the same inefficiencies as matrices, so they should be pre-allocated whenever possible
  - This mainly applies to adding in cells in a loop
One Practical Use of Cell Arrays:

- Store strings of different lengths in one variable
- Be careful about using () and {}
Cell Arrays: Referring to Cell Contents

• Referring to the contents of a cell may look intimidating
  • But it makes sense!

• Because cell arrays are numerically indexed
  • Can use loops to go through entries
Final Thoughts...

Cell Arrays are a little confusing at first, but they are necessary so we can...

- Mix strings of different lengths in a single variable
- Mix numbers, matrices, vectors, strings, characters in a single variable

Cell arrays are most commonly used in two situations:

- You want to store a list of words that are different lengths
- You want to read in a data file that has both numbers and words/letters
  - For this, we need to move onto “Advanced File I/O”