Matlab Programming

Outline

• Announcements:
  – Homework 1: due Wed. by 5, by e-mail
  – Remember: text & subject
  – Last day to add/drop or change credit/audit
• Iteration
• Conditionals and logic
• M-files

Iteration

• For loops in Matlab use index-notation:
  for j=st:step:en;
  <commands involving j> ;
  end
• Example: u'*v
  tot=0;
  for j=1:length(u);
    tot=tot+u(j)*v(j);
  end
**Conditionals**

- Conditional statements control flow of a program
  
  ```matlab
  if(logic);
  <commands executed if true>;
  else;
  <commands executed if false>;
  end
  ```

- Matlab also has switch:
  
  ```matlab
  switch(j);
  case a: <commands if a==j>;
  case b: <commands if b==j>;
  end
  ```

**Logic**

- Relational operators: \((R \times R \rightarrow B)\)
  
  - \(<, >, \leq, \geq, ==, \neq\)
  - isnan(a), isinf(a)

- Logical operators: \((B \times B \rightarrow B)\)
  
  - & (and), | (or), ~ (not), xor (which is just \(\neq\))

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>a&amp;b</th>
<th>a</th>
<th>b</th>
<th>xor(a,b)</th>
<th>a==b</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>0</td>
<td>T</td>
<td>0</td>
</tr>
<tr>
<td>T</td>
<td>0</td>
<td>0</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>0</td>
<td>T</td>
<td>0</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

- Matlab’s Boolean type is a logic array
  
  - Logical operators work with doubles, too
    - 0 => false; anything else => true
  
  - Logical operators are defined for arrays:
    
    - a=1:5; %[1 2 3 4 5]
    - b=a<3
      
      ```matlab
      b= 1 1 0 0 0
      ```
Logic--searching with find

- Find--searches for the truth (or not false)
  - \( b=\begin{bmatrix} 1 & 0 & -1 & 0 & 0 \end{bmatrix}; \)
  - \( I=\text{find}(b) \)
    - \( I=[1 \ 3] \quad b(I) \) is an array without zeros

Find for matrices

- \( M=\text{ones}(3,1)*\begin{bmatrix} 1 & 2 & 3 \end{bmatrix}; \)
  - \( I=\text{find}(M==2) \)
    - \( I=4 \ 5 \ 6 \)
  - \( [I,J]=\text{find}(M==2) \)
    - \( I=1 \ 2 \ 3 \)
    - \( J=2 \ 2 \ 2 \)

Programming

- Matlab programs are stored in “m-files”
  - Text files containing Matlab commands and ending with .m
  - Script m-files--commands executed as if entered on command line
  - Function m-files--analogous to subroutines or methods, maintain their own memory (workspace)
Scripts are Evil

- Scripts are EXTREMELY poor programming
  - Application specific, so difficult to reuse
  - Dangerous: variables created in scripts can overwrite existing variables
- If you send me a script, I will send it back!

Functions

- First line of file must be
  `function [outputs]=fname(inputs)`
  - outputs and inputs can have 0, 1, or many variables
  - Ex: `[U,V]=SSHvel(SSH, lat, lon)`
    - Requires 3 inputs (referred to in SSHvel.m as SSH, lat, lon)
    - Returns 2 outputs. Whatever U and V are when SSHvel finishes are returned to the workspace (or calling function)
  - `>> [locU,locV]=SSHvel(locSSH, loclat, loclon)`

Generic Function Format

1. Function statement
2. 1st comment—used by `lookfor`
3. Comments returned by `help SSHvel`—Should indicate how to call, and describe what it does
4. Error checking—check that sizes are correct/consistent. Return a helpful message with `error` command
5. Code
**Example: Geostrophy**

- We want to create a function to compute velocity from SSH data

```
SSH  \rightarrow  SSHvel.m  \rightarrow  U  \quad V
```

**Summary**

- Iteration with for (while exists too)
- Conditionals with if and switch
- Logical operators and vectorized
- 0=false
- find searches for truth
- Extend matlab with m-file functions
  - [output]=fname(input)