

NLAA Lab 1

Matlab catch-up

Julian Hall

School of Mathematics
University of Edinburgh

`jajhall@ed.ac.uk`

21 September 2016

Timing calculations in Matlab

Timing calculations in MATLAB: Wall clock

- The commands `tic` and `toc` allow the *elapsed* time of a MATLAB calculation to be determined
 - Think of `tic` as starting a stopwatch
 - The value of `toc` is the elapsed time since the previous `tic`
- For the following, `AinvTime` is the elapsed time since the call to `tic`
`tic;`
`Ainv = inv(A);`
`AinvTime = toc;`
- **But**, if these lines are typed into the command window then `AinvTime` includes the typing time!

What's in a name?

As with any MATLAB command, the value of `toc` can be assigned to a variable with *any* valid name but it is good practice to use a name which indicates what value is assigned to it

Timing calculations in MATLAB: Wall clock

Avoid counting the typing time by condensing

```
tic;
```

```
Ainv = inv(A);
```

```
AinvTime = toc;
```

into one line as

```
tic; Ainv = inv(A); AinvTime = toc;
```

so clock doesn't start until "Return" is pressed

Formatting data for output in Matlab

Formatting data for output in MATLAB: Why?

- Many MATLAB commands generate output
- Output is “anonymous” and can take up lots of space
 - Without “LHS =” command, just get ans = followed by data
 - Example: `>> size([1 2 3;4 5 6])` yields
ans =
2 3
 - With “LHS =” command, things are little better
 - Example: `>> [m, n] = size([1 2 3;4 5 6])` yields
m =
2
n =
3
- Preferable to suppress printing of the output with “;” and format it clearly

Formatting data for output in MATLAB: `sprintf`

`sprintf`

`sprintf` allows a formatted string to be “constructed” from components including

- string
- integer
- real

Formally

```
str = sprintf(formatSpec,A1,...,An)
```

formats the data in arrays $A_1, \dots, A_n$ according to `formatSpec` and returns the results to string `str`

Formatting data for output in MATLAB: sprintf

Giving sprintf a single string

```
>> sprintf('This is a string')
```

yields

```
ans =
```

```
This is a string
```

Formatting data for output in MATLAB: sprintf

Inserting an integer into a string

- Indicate the position of the integer in **formatSpec** using %d
- The integer value to use must be defined after formatSpec
- Example:

```
>> n = 10; A = rand(n);  
>> sprintf('The dimension of A is %d', n)  
yields  
ans =  
The dimension of A is 10
```

Formatting data for output in MATLAB: sprintf

Inserting a real into a string

- Indicate the position of the real in formatSpec using %g
- The real value to use must be defined after formatSpec
- Example:

```
>> n = 10; A = rand(n);  
>> tic; Ainv = inv(A); AinvTime = toc;  
>> sprintf('The time for inv(A) is %g', AinvTime)  
yields  
ans =  
The time for inv(A) is 7.2e-05
```

Formatting data for output in MATLAB: sprintf

Inserting an integer and a real into a string

- Indicate the position of the integer and real using %d and %g
- The values to use must be defined *in order* after formatSpec
- Example:

```
>> n = 10; A = rand(n);  
>> tic; Ainv = inv(A); AinvTime = toc;  
>> sprintf('For n = %d the time for inv(A) is %g', n, AinvTime)  
yields  
ans =  
For n = 10 the time for inv(A) is 7.2e-05
```

Formatting data for output in MATLAB: `sprintf`

Using `disp` to save space

- Single output statement still uses several lines
- Avoid this by writing output to a string which is then displayed using **`disp`**
- Example:

```
>> str = sprintf('For n = %d the time for inv(A) is %g', n,  
AinvTime); disp(str)
```

yields

```
For n = 10 the time for inv(A) is 7.2e-05
```

Formatting data for output in MATLAB: `sprintf`

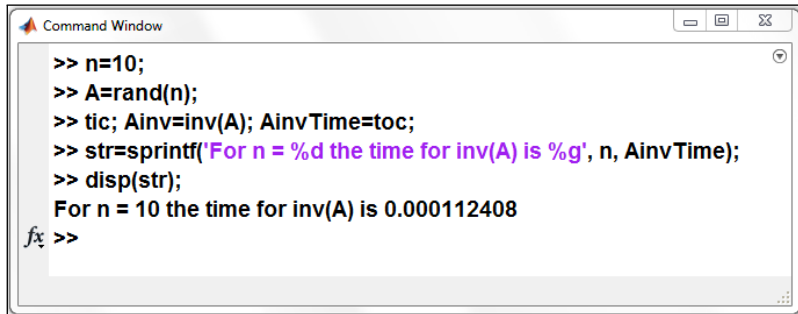
Further formatting

- Much more sophisticated formatting of output is possible
- Aim for clear output, aligned in columns
 - Makes trends easier to spot
 - Reduce the scope for error
- Can also write output to files for further processing using Excel, for example

Matlab scripts

- MATLAB scripts allow sequences of commands to be stored and performed
- Allow systematic scientific study
- Stepping stone to programming with MATLAB

MATLAB: Scripts - experiments with random matrix

A screenshot of the MATLAB Command Window. The window has a title bar with the MATLAB logo and the text "Command Window". Inside the window, the following commands are entered:

```
>> n=10;  
>> A=rand(n);  
>> tic; Ainv=inv(A); AinvTime=toc;  
>> str=sprintf('For n = %d the time for inv(A) is %g', n, AinvTime);  
>> disp(str);
```

The output of the last command is displayed as:

```
For n = 10 the time for inv(A) is 0.000112408
```

Below the output, there is a prompt character *fx* followed by

```
>>
```

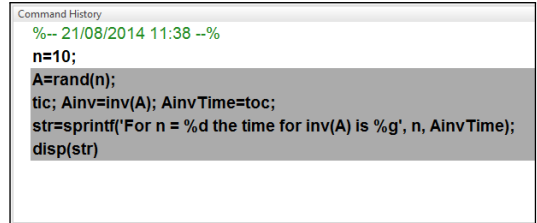
.

```
fx >>
```

- **Aim:** To repeat the experiment with different values of n
- Want to avoid having to re-type all the commands
- Collect the commands into a **script**

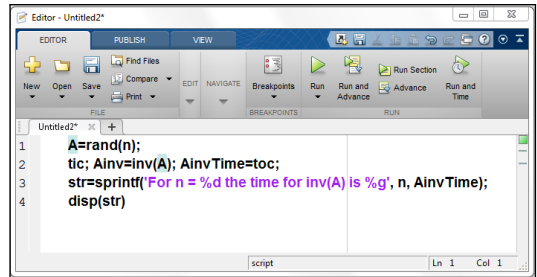
MATLAB: Creating a script

- Open the Command History
 - Select the commands of interest
 - Right-click and select **Create Script**
-
- Opens the **Editor Window**



The Command History window displays a list of executed commands. The first command is a date and time stamp: `%-- 21/08/2014 11:38 --%`. The subsequent commands are: `n=10;`, `A=rand(n);`, `tic; Ainv=inv(A); AinvTime=toc;`, and `str=sprintf('For n = %d the time for inv(A) is %g', n, AinvTime); disp(str)`. The last command is highlighted with a grey background.

```
Command History
%-- 21/08/2014 11:38 --%
n=10;
A=rand(n);
tic; Ainv=inv(A); AinvTime=toc;
str=sprintf('For n = %d the time for inv(A) is %g', n, AinvTime);
disp(str)
```



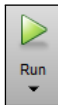
The Editor window shows a script file named 'Untitled2*'. The script contains the following code:

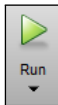
```
1 A=rand(n);
2 tic; Ainv=inv(A); AinvTime=toc;
3 str=sprintf('For n = %d the time for inv(A) is %g', n, AinvTime);
4 disp(str)
```

The script is displayed in a window with a toolbar at the top containing icons for File, Edit, Navigate, Breakpoints, and Run. The status bar at the bottom indicates 'script' and 'Ln 1 Col 1'.

MATLAB: Running a script

Run scripts in two ways



- Click on  to (save and) run the script
- Type the name of the script in the Command Window

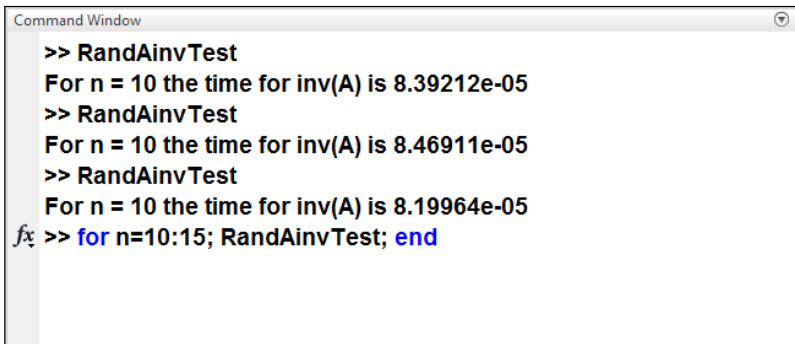
```
Command Window
>> RandAinvTest
For n = 10 the time for inv(A) is 8.39212e-05
>> RandAinvTest
For n = 10 the time for inv(A) is 8.46911e-05
>> RandAinvTest
For n = 10 the time for inv(A) is 8.19964e-05
fx >>
```

MATLAB: Loops

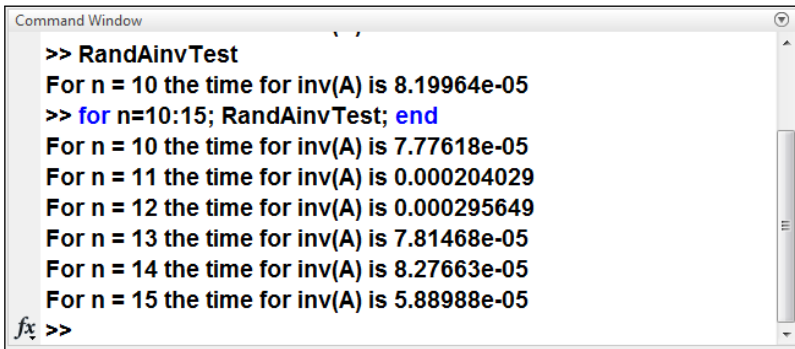
- Run the script for a set of values of n by using a **for loop**

```
for n=10:15; RandAinvTest; end
```

- Tells MATLAB to run the script with $n = 10, 11, \dots, 15$

A screenshot of the MATLAB Command Window. The window has a title bar that says "Command Window" and a small icon on the right. The text inside the window shows three manual executions of the "RandAinvTest" script, each followed by a message indicating the time taken for the inversion of matrix A for n=10. The times are 8.39212e-05, 8.46911e-05, and 8.19964e-05. The final line shows the execution of a for loop that runs "RandAinvTest" for n from 10 to 15. A cursor icon is visible at the end of the last line.

```
>> RandAinvTest
For n = 10 the time for inv(A) is 8.39212e-05
>> RandAinvTest
For n = 10 the time for inv(A) is 8.46911e-05
>> RandAinvTest
For n = 10 the time for inv(A) is 8.19964e-05
fx >> for n=10:15; RandAinvTest; end
```

A screenshot of the MATLAB Command Window. The window has a title bar that says "Command Window". Inside, the following text is displayed: ">> RandAinvTest", "For n = 10 the time for inv(A) is 8.19964e-05", ">> for n=10:15; RandAinvTest; end", "For n = 10 the time for inv(A) is 7.77618e-05", "For n = 11 the time for inv(A) is 0.000204029", "For n = 12 the time for inv(A) is 0.000295649", "For n = 13 the time for inv(A) is 7.81468e-05", "For n = 14 the time for inv(A) is 8.27663e-05", "For n = 15 the time for inv(A) is 5.88988e-05", and finally "fx >>". The window includes a scrollbar on the right side.

```
>> RandAinvTest
For n = 10 the time for inv(A) is 8.19964e-05
>> for n=10:15; RandAinvTest; end
For n = 10 the time for inv(A) is 7.77618e-05
For n = 11 the time for inv(A) is 0.000204029
For n = 12 the time for inv(A) is 0.000295649
For n = 13 the time for inv(A) is 7.81468e-05
For n = 14 the time for inv(A) is 8.27663e-05
For n = 15 the time for inv(A) is 5.88988e-05
fx >>
```