

# CFCS1

## Matlab Overview

Miles Osborne

School of Informatics  
University of Edinburgh  
`miles@inf.ed.ac.uk`

January 7, 2008

1 What is Matlab?

2 Examples

# What is Matlab?

Matlab is a *task-specific* programming language and environment:

- Tailored to linear algebra and statistical operations.
- Allows for visualisation of results.
- Commercial, closed-domain system.

# History

- MARTix LABoratory was invented in late 70s.
  - It was initially designed for students to solve linear algebra problems without having to learn Fortran.
- It is now commercially developed by *The MathWorks*.
- An Open Source version exists (*octave*).

See: <http://en.wikipedia.org/wiki/Matlab>

# Overview

Matlab consists of a programming language and an environment:

- The programming language is called *M-code* (or *M*).
- The environment is called the *Desktop*.

Matlab has a core library and a set of (optional) extensions:

- Image Processing.
- Compiler.
- Statistics.
- Financial Modelling.
- Etc.

# Motivation for using Matlab

Matlab is a complete programming language:

- Anything we can do in Java (or ...) we can do in Matlab.
- (Which means, anything we can do in Matlab, we can do in our favourite language).
- Why invent a new language?

# Motivation for using Matlab

Computer Science has a long history of creating special-purpose languages:

- Yacc and Lex are tiny languages for writing compilers.
- Latex is a language for typesetting.
- Bash, csh etc are languages for running Unix commands.
- SQL is used for databases.

# Motivation for using Matlab

Special purpose languages tradeoff high-level abstraction for speed:

- Matlab allows for rapid creation of mathematical models.
- People's productivity is improved.
  - We do not have to reinvent common mathematical operations.
  - We can be confident that these operations are wcorrect.
- But, the price is execution speed and generality.

Frequently, people are expensive and machines are cheap:

Use the *right tool for the job*



# To run

To run *Matlab*, from a Dice Box:

```
[beatles]miles: matlab
```

(To exit, use the menu option *Exit* under the file menu)

## Example 1

## Matrix Multiplication

$$A = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} \quad B = \begin{pmatrix} 3 & 4 \\ 5 & 6 \end{pmatrix}$$

$$A \cdot B = \begin{pmatrix} 11 & 14 \\ 13 & 16 \end{pmatrix}$$

# Example 1

Matlab version:

```
>> A = [2 1; 1 2]; B = [3 4; 5 6];
```

```
>> A * B
```

```
ans =
```

```
11    14
```

```
13    16
```

```
>>
```

# Example 1

Notice:

- We have defined two *variables* ( $A$  and  $B$ ).
- Each variable has a *type* (here, they are matrices).
- Commands are entered into the *interpreter*.
  - We can also define our own programs and load them in.
- Matlab correctly multiplies two matrices.

## Example 2

### Computing a weighted average

Age	Number of people
1	1
2	1
3	2

Weighted average is 2.25

## Example 2

### Computing a weighted average

$$\bar{x} = \frac{\sum_{j=1}^N x_j N(x_j)}{\sum_{j=1}^N N(x_j)}$$

- Assume we have an array of data  $x$ .
- Assume we have an array  $N$  of item  $x_i$  frequencies.

## Example 2

```
function ave = myaverage(x,N)
% compute the weighted average of vector x
% weighted by vector N
sizex = size(x);
sizeN = size(N);
if sizex(2) ~= sizeN(2)
    disp('Error: Arrays must agree!')
else
    s = x.*N;
    total = sum(N);
    ave = sum(s)/total;
end
```

## Example 2

Here we have defined a *function*:

- We have added comments.
- We check the array dimensions for legality.
- The total is computed using a *sum* operation.
  - There is no looping.
- We also use array operations (a special multiplication).
- The function returns a *scalar*.



## Example 3

- Matlab supports *numerical optimisation*
- Numerical optimisation deals with finding the best solution to some problem.
- Most Cognitive Science tasks can be considered as numerical optimisation.

### Modelling ambiguity

Consider the sentence:

*I read the book on the atom*

What does this mean? What could it mean?

## Example 3

### Computing a weighted average

Suppose we have a function:

$$f(x) = x^2$$

What is the smallest  $x$  value in a given range of  $f(x)$  values?

## Example 3

```
>> f = @(x)x.^2;  
>> z = fminbnd(f, 1, 2)
```

```
z =
```

```
1.001
```

## Example 3

- We have defined an anonymous function  $f$ , taking a single argument.
- That function is an argument to an *optimisation* function.
- We have searched in the range of possible  $y$  values from 1.001 to 2.
  - (The boundary numbers are not considered).
- ... and have found the smallest value.

# Summary

- Matlab is a high level programming language and environment.
- It is well suited to Cognitive Science.
- We looked at a range of examples hinting at the power of Matlab.