CFCS1

Vectors in MATLAB

Miles Osborne

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

January 17, 2008

Overview

A MATLAB vector is a one-dimensional array of the same type:

Vector Examples

[3 9 6]	Three integer elements
[ ]	No elements
[1.2 0.3]	Two real number elements

(There are no differences between vectors and matrices, apart from the dimensions)

Overview

Vectors are created either as a by-product of some operation or directly:

Vector Creation

```
a = [3 9 6]   %Creates a vector called a
b =  a * 3    %Creates a new vector
```

Note that there is no need to specify the size of the vector.

Overview

Individual components are accessed using indexing:

Vector Creation

```
a = [3 9 6]   %Creates a vector called a
c = a(1)      % Select the first element
```

- Round-brackets are used to specify an entry.
- Indexing starts from 1 (not zero).

Overview

Sequence components are specified using the colon notation:

Vector Creation

```
a = [3 9 6]   %Creates a vector called a
c = a(1:2)    % Create the vector [3 9]
```


Overview

Items can be directly replaced:

Vector Creation

```
a = [3 9 6]   %Creates a vector called a
a(2) = 8      % Create the vector [3 8 6]
a(5) = 1      % Create the longer vector [3 8 6 0 1]
```

Note: implied entries are set to zero.

Linear Algebra Operations

MATLAB directly supports common operations:

Linear Algebra Examples

```
norm(a)                          % Find the norm (length) of a
dot(a, b)                        % Dot product of a and b
a * 4                            % Scalar multiplication
a - b                            % Vector subtraction
a + b                            % Vector addition
a / norm(a)                      % Create a unit vector
dot(a,b) / (norm(a) * norm(b))   % Cosine angle
```

Useful Operations

MATLAB has a library with many common operations:

Useful Operations

sum(**a**)	$\sum_{i=1}^{n} a_i$
prod(**a**)	$\prod_{i=1}^{n} a_i$
sort(**a**)	% Sort the vector
max(**a**)	% Find the largest element
min(**a**)	% Find the smallest element
length(**a**)	%How many elements in the vector

In general, before writing some code, see if MATLAB already supports it!

Quiz

A *stack* is a data structure supporting the following operations:

- Push: add an element to a list (eg push(a,(b c)) = (a b c)).
- Pop: remove the first element from the list (eg pop((a b c)) = (b c))

How can MATLAB implement these operations?

Vectorising

- MATLAB is a general-purpose programming language
- It also has efficient support for common operations over vectors.
- *Vectorising* means using these operations in place of explicit control constructs.

Vectorising : An Example

- *Logical* indexing uses a vector to specify whether a corresponding component in another vector should be extracted.
- A one in that vector means extract the corresponding element (and a zero means ignore it).

Logical Indexing

a = [1 2 3]
b = [0 1 1]
a(logical(**b**)) = [2 3]

Vectorising : Another example

- The operation $>$ tests whether each element in a vector is greater than some number.
- Likewise, the operation $<$ tests whether each element in a vector is less than some number.
- How can we use vectorising to remove numbers which fall outside of some range?

Trimming bad values

$\mathbf{a} = [-999\ 2\ 3\ 3\ 999]$
$\mathbf{b} = \mathrm{a} > 0\ \&\ a\ <\ 5$
$\mathbf{a}(\mathrm{logical}(\mathbf{b})) = [2\ 3\ 3]$

Vectorising : Another example

- The operation $==$ tests whether each element in a vector is equal to some number.
- How can we use vectorising to count the number of components that are equal to some number?

Trimming bad values

$\mathbf{a} = [-999\ 2\ 3\ 3\ 999]$
$\mathrm{sum}((\mathbf{a} == 3)) = 2$

Note: this does not use *logical indexing*, we simply count the number of true elements in the logical vector.

Summary

- Vectors in MATLAB are first-class types.
- There are a rich variety of operations over them.
- Vectorising is a technique for writing faster, more compact programs.