

# Vectorized Logical Operations

### Vectorized code to support plotting

$$f(x) = \frac{\sin(5x)\exp(-x/2)}{1+x^2} \quad \text{for } -2 \leq x \leq 3$$

```

x = linspace(-2,3,200);
y = sin(5*x).*exp(-x/2)./(1 + x.^2);
plot(x,y)

```

### Vectorized code to support plotting

$$f(x) = \frac{\sin(5x)\exp(-x/2)}{1+x^2} \quad \text{for } -2 \leq x \leq 3$$

```

x = linspace(-2,3,200);
y = sin(5*x).*exp(-x/2)./(1 + x.^2);
plot(x,y)

```

Element-by-element arithmetic operations on arrays

### Zeroing out values below a threshold

```

y= ... % sound data
T= 0.08; % threshold
for k= 1:length(y)
    if abs(y(k))<T
        y(k)= 0;
    end
end

```

```

y= ... % sound data
T= 0.08; % threshold
y= y .* abs(y)>T;

```

← vectorized version

### Extracting values from a vector

Extract all the values in (-1,10)

```

v = [-2 17 3 1.2 11 3 4 -2 .2 0 3.1 4 -12 -2 9]
w = [];
for k= 1:length(v)
    if v(k)>-1 && v(k)<10
        w = [w v(k)];
    end
end

```

vectorized version → `w = v( v>-1 & v<10 );`

### Extracting values from a vector

Extract all the values in (-1,10)

```

v = [-2 17 3 1.2 11 3 4 -2 .2 0 3.1 4 -12 -2 9]
w = v( v>-1 & v<10 );

```

[0 1 1 0 1 1 0 1 1 1 1 0 1 1]

### Extracting values from a vector

v [-2 17 3 1.2 11 3 4 -2 .2 0 3.1 4 -12 -2 9]

Extract all the values in (-1,10)

```
w = v( v > -1 & v < 10 );
```

v( )

Indices (positive integers) go here ... usually.

### Extracting values from a vector

v [-2 17 3 1.2 11 3 4 -2 .2 0 3.1 4 -12 -2 9]

Extract all the values in (-1,10)

```
w = v( v > -1 & v < 10 );
```

v( [0 0 1 1 0 ... 1 0 1 1] )

If these were plain (numeric) 0s and 1s, then this would not work! BUT these 0s and 1s come from logical operations, and therefore have the logical property.

Matlab allows the use of logical values instead of indices for accessing particular cells in a vector. The length of the logical array must be the same as the length of the vector.

### Extracting values from a matrix

m [ 3 4 -2 .2 0  
0.1 17 3 1.2 11  
3.1 4 -12 -2 9  
-9 -1.1 -3 0 8 ]

Extract all the values in (1,10)

```
w = m( m > 1 & m < 10 );
```

Bitwise (non-short-circuiting) operator

w [ 3  
3.1  
4  
4  
3  
9  
8 ]