Overview

Vectorization is the use of MATLAB’s implementation of matrix algebra syntax or array operators to perform calculation without the explicit use of loops.

Vectorized expression:  
\[
x = \text{linspace}(0,2\pi);
\]
\[
y = \sin(x);
\]
Because \(x\) is a vector, MATLAB automatically creates \(y\) as a vector of the same shape. Each element of \(y\) is the sine of the corresponding element of \(x\)

Equivalent Loop:  
\[
n = 100;
\]
\[
dx = 2\pi/(n-1);
\]
\[
x(1) = 0;
\]
\[
y(1) = \sin(x(1));
\]
for \(i=2:n\)
\[
x(i) = x(i-1) + dx;
\]
\[
y(i) = \sin(x(i));
\]
end

Advantages

Vectorization is good because

- Vectorization enables writing of code that is compact and idiomatic.
- Compact, idiomatic code is easier to read and debug.
- Vectorized code is faster, even though the same computations are performed.

Matrix Operations are Vectorized

The MATLAB *, +, and − operators adhere (mostly) to the rules of linear algebra.

Examples:
\[
\begin{align*}
\text{>> } x &= [1; 2; 3]; \quad y = [5; 1; -2]; \\
\text{>> } z &= x + y \\
&= 6 \\
&= 3 \\
&= 1 \\
\text{>> } A &= [2 -1 3; 4 0 7; 5 9 -6]; \\
\text{>> } u &= A*x \\
&= 9 \\
&= 25 \\
&= 5 \\
\end{align*}
\]

Scalar addition

You cannot add a scalar to a vector or a matrix, but MATLAB allows the following abuse of the notation of linear algebra.
\[
\begin{align*}
\text{>> } s &= 2 \\
&s = 2 \\
\text{>> } B &= A + s \\
&B =
\begin{bmatrix}
4 & 1 & 5 \\
6 & 2 & 9 \\
7 & 11 & -4
\end{bmatrix}
\end{align*}
\]
Array Operators

There are situations where vectorization would be good, but not supported by the rules of linear algebra.

Example: Compute the area of a set of circles, $a = \pi r^2$, where $r$ is a vector of radii. According to the rules of linear algebra, only square matrices can be squared.

To help the programmer, without breaking the rules of linear algebra, MATLAB provides array operators. In the case of the square (or any power), the expression $y = x^n$ creates a vector $y$ of the same shape as $x$, and each element of $y$ is the square of corresponding element of $x$.

<table>
<thead>
<tr>
<th>Vectorized expression:</th>
<th>Equivalent Loop:</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a = \pi r^2$;</td>
<td>for $i=1:length(r)$</td>
</tr>
<tr>
<td></td>
<td>$a(i) = \pi x(i)^2$;</td>
</tr>
<tr>
<td></td>
<td>end</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Operator</th>
<th>Meaning</th>
<th>Vectorized Example</th>
<th>Equivalent Loop</th>
</tr>
</thead>
<tbody>
<tr>
<td>.*</td>
<td>Element-by-element multiplication</td>
<td>$z = x.*y$</td>
<td>for $i=1:length(x)$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$z(i) = x(i)*y(i)$;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>end</td>
</tr>
<tr>
<td>./</td>
<td>Element-by-element division</td>
<td>$z = x./y$</td>
<td>for $i=1:length(x)$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$z(i) = x(i)/y(i)$;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>end</td>
</tr>
<tr>
<td>.^</td>
<td>Raise each element to a power</td>
<td>$z = x.^(1/3)$</td>
<td>for $i=1:length(x)$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$z(i) = x(i)^(1/3)$;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>end</td>
</tr>
</tbody>
</table>

Note: There is no need for .+ , .- operators.