

# Including MATLAB Code into L<sup>A</sup>T<sub>E</sub>X Documents

Gerald Recktenwald\*

May 18, 2011

## Overview

This document shows how to use the `matlab` package to incorporate MATLAB source code into a L<sup>A</sup>T<sub>E</sub>X document. Short sections of code formatted with the `mtext` environment like this

L<sup>A</sup>T<sub>E</sub>X code:

```
\begin{mtext}
x = 1;
y = 3;
z = x/y;
\end{mtext}
```

Typeset appearance

```
x = 1;
y = 3;
z = x/y;
```

Code in external files is imported and typeset with one of these three commands: `VerbListing`, `VerbListingBoxed`, and `VerbListingNumber`. Those commands are often used inside a `Listing` environment like this:

```
\begin{Listing}
\VerbListingBoxed{linetest.m}
\caption{The \texttt{lintest} function in a floating ‘‘Listing’’ environment.}
\label{mfile:linetest}
\end{Listing}
```

Those commands produce Listing 1 on page 2.

## Example

Consider fitting a line to the four points (1,1), (2,2), (4,2), and (5,3). The `linefit` function does all the work once the  $(x,y)$  data are specified.

```
>> x = [1 2 4 5];    y = [1 2 2 3];
>> c = linefit(x,y)
c =
    0.4000
    0.8000
```

---

\*Mechanical and Materials Engineering Department, Portland State University, Portland, OR, 97201, [gerry@me.pdx.edu](mailto:gerry@me.pdx.edu)

```

function [c,R2] = linefit(x,y)
% linefit    Least-squares fit of data to y = c(1)*x + c(2)
%
% Synopsis:  c      = linefit(x,y)
%            [c,R2] = linefit(x,y)
%
% Input:     x,y = vectors of independent and dependent variables
%
% Output:    c = vector of slope, c(1), and intercept, c(2) of least sq. line fit
%            R2 = (optional) coefficient of determination; 0 <= R2 <= 1
%            R2 close to 1 indicates a strong relationship between y and x
if length(y)~= length(x), error('x and y are not compatible'); end

x = x(:); y = y(:); % Make sure that x and y are column vectors
A = [x ones(size(x))]; % m-by-n matrix of overdetermined system
c = (A'*A)\(A'*y); % Solve normal equations
if nargin>1
    r = y - A*c;
    R2 = 1 - (norm(r)/norm(y-mean(y)))^2;
end

```

Listing 1: The `linefit` function in a floating “Listing” environment where the code is formatted with the `VerbListingBoxed` function.

A qualitative check on the success of the fit is obtained by plotting the equation of the fit along with the original data. The following statements produce the plot shown in Figure 1:

```

>> xfit = [0 6]; % Evaluate fit over this range of x
>> yfit = c(1)*xfit + c(2); % Values of the fit function
>> plot(x,y,'o',xfit,yfit,'-')
>> grid on;
>> xlabel('x values');
>> ylabel('y data and fit function');

```

---

```

function [c,R2] = linefit(x,y)
% linefit    Least-squares fit of data to y = c(1)*x + c(2)
%
% Synopsis:   c      = linefit(x,y)
%             [c,R2] = linefit(x,y)
%
% Input:      x,y = vectors of independent and dependent variables
%
% Output:     c = vector of slope, c(1), and intercept, c(2) of least sq. line fit
%             R2 = (optional) coefficient of determination; 0 <= R2 <= 1
%             R2 close to 1 indicates a strong relationship between y and x
if length(y)~= length(x), error('x and y are not compatible'); end

x = x(:); y = y(:); % Make sure that x and y are column vectors
A = [x ones(size(x))]; % m-by-n matrix of overdetermined system
c = (A'*A)\(A'*y); % Solve normal equations
if nargout>1
    r = y - A*c;
    R2 = 1 - (norm(r)/norm(y-mean(y)))^2;
end

```

---

Listing 2: The `linefit` function in a floating “Listing” environment where the code is formatted with the `VerbListing` function.

---

```

1 function [c,R2] = linefit(x,y)
2 % linefit    Least-squares fit of data to y = c(1)*x + c(2)
3 %
4 % Synopsis:   c      = linefit(x,y)
5 %             [c,R2] = linefit(x,y)
6 %
7 % Input:      x,y = vectors of independent and dependent variables
8 %
9 % Output:     c = vector of slope, c(1), and intercept, c(2) of least sq. line fit
10 %            R2 = (optional) coefficient of determination; 0 <= R2 <= 1
11 %            R2 close to 1 indicates a strong relationship between y and x
12 if length(y)~= length(x), error('x and y are not compatible'); end
13
14 x = x(:); y = y(:); % Make sure that x and y are column vectors
15 A = [x ones(size(x))]; % m-by-n matrix of overdetermined system
16 c = (A'*A)\(A'*y); % Solve normal equations
17 if nargout>1
18     r = y - A*c;
19     R2 = 1 - (norm(r)/norm(y-mean(y)))^2;
20 end

```

---

Listing 3: The `linefit` function in a floating “Listing” environment where the code is formatted with the `VerbListingNumber` function.

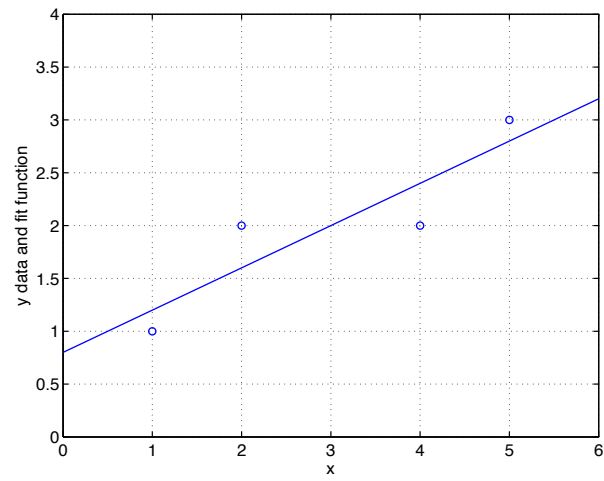


Figure 1: Least-squares fit of four data points to a line.