Solutions

A fitting problem and a monotonic interpolant alternative

The following script solves both problems:

```
x = [-1.5 -1.25 -1.0 -0.75 -0.5 -0.25 0.0 0.25 0.5 0.75 1.0 1.25
1.5];
y = [0.0 0.25 0.5 1.25 3.0 6.0 8.0 6.0 3.0 1.25 0.5 0.25 0.0];

plot(x,y,'+')
xlabel('x')
ylabel('y','Rotation',0)

hold on

n = length(x); % n+1 in question
m = ones(n);
m(:,2)=x';
for i = 3:n
    m(:,i) = x.^(i-1)';
end

m = fliplr(m);
[mm, ipiv, a, info] = f07aa(m, y');

% The coefficients ...
a'
% ... and using MATLAB's polyfit
p = polyfit(x,y,n-1)

% Evaluate the polynomial over a range of values
range = x(1):0.05:x(end);
z = polyval(a,range);

plot(range,z,'-g')

% Now with e01be/f
[d, ifail] = e01be(x, y);
[pf, ifail] = e01bf(x, y, d, range);

plot(range,pf,'-r')

legend('Data','Vandermonde','E01BE')

hold off
```