**Implementing data fitting in MATLAB**

See codes posted under today's date on website, and Ch. 3 of 301.pdf!!!!!

**load data**

```
% Load the data
load linefit.dat
x=linefit(:,1);
y=linefit(:,2);
figure(1)
set(gca,'FontSize',20)
plot(x,y,'O')
legend('Raw Data')
```

**Define a list of points at which you'd like to plot the fitted function**

```
% These are the points we would like to interpolate to
xp=0:0.1:7;
```

**Get polynomial coefficients for least-squares line, $p_1(x) = a_1 x + a_0$**

```
% Polynomial fits -- LEAST SQUARES
clin=polyfit(x,y,1);

clin =   0.830757023627601   2.138914031581835
```

First entry is $a_1$, second is $a_0$

**Use these coefficients to get the list of interpolated values yp**

```
yp=polyval(clin,xp);
```

**Repeat the process for an order 2 polynomial least-squares fit, and plot.**

```
clin2=polyfit(x,y,2);
yp2=polyval(clin2,xp);

figure(2),
set(gca,'FontSize',20)
plot(x,y,'O', ...
    xp,yp,'m',xp,yp2,'g')
legend('Raw Data','Least Squares Line','Least Squares 2nd order polyn
```

**Use polyinterp for polynomial interpolation that crosses through all N DATAPOINTS, VIA LAGRANGE POLYNOMIALS**

```
ypn=polyinterp(x,y,xp);
figure(3),
set(gca,'FontSize',20)
plot(x,y,'O',xp,ypn,'m')
legend('Raw Data','Poly interpolation via Lagrange')
```

**Compute root-mean-squared error for the Least-squares line and least-squares order 2 polynomial fit**

```
% Error
E_2_1 = sqrt( 1/length(x) * sum( abs(y-polyval(clin,x)).^2) )
E_2_2 = sqrt( 1/length(x) * sum( abs(y-polyval(clin2,x)).^2) )
```

## Piecewise interpolation

```
yint=interp1(x,y,xp); % Linear functions
yspline=spline(x,y,xp); % Cubic functions (splines)
figure(4),
set(gca,'FontSize',20)
plot(x,y,'O',...
    xp,yint,'m',xp,yspline,'k')
legend('Raw Data','Linear Interpolation','Spline interpolation')
```

**Nonlinear least-squares fit, to a gaussian function.**

**First, plot data**

```
load gaussfit.dat
x2=gaussfit(:,1) ;
y2=gaussfit(:,2) ;
figure(5),
set(gca,'FontSize',20)
plot(x2,y2,'O')
legend('Raw Data')
```

**Define a list of points at which you'd like to plot the fitted function**

```
xga=-3:0.1:3;
```

**Write a function gafit.m that computes the summed error as a function of the unknown coefficients $A$ and $B$ in the function $f(x) = A\exp(-Bx^2)$**

(note, this is not the same as root-mean-squared ($E_2$) error, but minimizing it has the same effect, as discussed in class)

Parameters A and B are grouped into a row vector x0 here

```
function y=gafit(x0)

load gaussfit.dat
x=gaussfit(:,1);
y=gaussfit(:,2);

sum=0;
for j=1:length(x)
   sum=sum+ ( x0(1)*exp(-x0(2)*x(j)^2) ...
      -y(j)  )^2 ;
end
y=sum;
```

**Find the A and B that minimize this error, plot the resulting least-squares fit**

```
coeff=fminsearch('gafit',[1 1]);
a=coeff(1); b=coeff(2) ;
yga=a*exp(-b*xga.^2);
figure(6),
set(gca,'FontSize',20)
plot(x2,y2,'O',xga,yga,'m')
legend('Raw Data','Gaussian nonlinear least-squares fit')
```