```
% nlin_fit_kinetics_setup.m
%
% This MATLAB m-file sets up the problem of
% fitting the rate constants for the network
% of two reactions :
%    A + B --> C
%    C + B --> D
%
% The two parameters to be fixed are the rate
% constants for each reaction.
%
% The data to be fit are the concentrations of
% species A, C, and D for various times for
% several experiments of initial concentrations
% of A and B.
%
% K. Beers
% MIT ChE
% 12/9/2001

% First, we set the true values of the rate constants.
k1 = 1;
k2 = 10;

% We now set the experimental conditions.
Param_fix.num_experiments = 3;
Param_fix.conc_init = zeros(3,4);
Param_fix.num_time_data = zeros(3,1);
Param_fix.time_data = zeros(3,8);
% experiment # 1
Param_fix.conc_init(1,:) = [1 1 0 0];
Param_fix.num_time_data(1) = 8;
Param_fix.time_data(1,1:8) = [0 0.2 0.4 0.6 0.8 1.0 1.2 1.4];
% experiment # 2
Param_fix.conc_init(2,:) = [1 2 0 0];
Param_fix.num_time_data(2) = 6;
Param_fix.time_data(2,1:6) = [0 0.2 0.4 0.6 0.8 1.0];
% experiment # 3
Param_fix.conc_init(3,:) = [1 3 0 0];
Param_fix.num_time_data(3) = 6;
Param_fix.time_data(3,1:6) = [0 0.2 0.4 0.6 0.8 1.0];

% We now generate the data for each experiment using
% the true values of the rate constants.
theta_true = [k1; k2];
y = nlin_fit_kinetics(theta_true,Param_fix);

% To this we add a random noise according to a
% normal distribution.
```

```
std_dev = 0.01;
y = y + std_dev*randn(length(y),1);

% We now call the nonlinear parameter estimation
% routine for this data, using the true values
% as the initial guess.
calc_yhat = 'nlin_fit_kinetics';
theta_guess = theta_true;
alpha_CI = 0.05;
Options.atol = 1e-6;
Options.verbose = 2;
[theta,theta_CI,yhat,yhat_CI,Stats,iflag] = ...
   simple_nonlinear_LS(...
   y,calc_yhat,theta_guess,alpha_CI,Options,Param_fix);


% We now plot the experimental data along with the
% model predictions and the confidence intervals.
count_master = 0;
Afig = figure;
Cfig = figure;
Dfig = figure;
for iexp=1:3
   count = count_master;
   if(iexp == 1)
      symbol = 'o';
   elseif(iexp == 2)
      symbol = 's';
   else
      symbol = 'd';
   end
   % plot experimental data
   time_vect = Param_fix.time_data(iexp,1:Param_fix.num_time_data(iexp))';
   A_data = y(count+1:count+Param_fix.num_time_data(iexp));
   count = count + Param_fix.num_time_data(iexp);
   C_data = y(count+1:count+Param_fix.num_time_data(iexp));
   count = count + Param_fix.num_time_data(iexp);
   D_data = y(count+1:count+Param_fix.num_time_data(iexp));
   figure(Afig);
   plot(time_vect,A_data,symbol);
   figure(Cfig);
   plot(time_vect,C_data,symbol);
   figure(Dfig);
   plot(time_vect,D_data,symbol);
   % plot model predictions
   count = count_master;
   A_data = yhat(count+1:count+Param_fix.num_time_data(iexp));
   count = count + Param_fix.num_time_data(iexp);
   C_data = yhat(count+1:count+Param_fix.num_time_data(iexp));
   count = count + Param_fix.num_time_data(iexp);
   D_data = yhat(count+1:count+Param_fix.num_time_data(iexp));
```

```
    figure(Afig); hold on;
    plot(time_vect,A_data);
    figure(Cfig); hold on;
    plot(time_vect,C_data);
    figure(Dfig); hold on;
    plot(time_vect,D_data);
    % plot lower CI interval on predictions
    count = count_master;
    A_data = yhat(count+1:count+Param_fix.num_time_data(iexp)) - ...
        yhat_CI(count+1:count+Param_fix.num_time_data(iexp));
    count = count + Param_fix.num_time_data(iexp);
    C_data = yhat(count+1:count+Param_fix.num_time_data(iexp)) - ...
        yhat_CI(count+1:count+Param_fix.num_time_data(iexp));
    count = count + Param_fix.num_time_data(iexp);
    D_data = yhat(count+1:count+Param_fix.num_time_data(iexp)) - ...
        yhat_CI(count+1:count+Param_fix.num_time_data(iexp));
    figure(Afig); hold on;
    plot(time_vect,A_data,'-.');
    figure(Cfig); hold on;
    plot(time_vect,C_data,'-.');
    figure(Dfig); hold on;
    plot(time_vect,D_data,'-.');
    % plot upper CI interval on predictions
    count = count_master;
    A_data = yhat(count+1:count+Param_fix.num_time_data(iexp)) + ...
        yhat_CI(count+1:count+Param_fix.num_time_data(iexp));
    count = count + Param_fix.num_time_data(iexp);
    C_data = yhat(count+1:count+Param_fix.num_time_data(iexp)) + ...
        yhat_CI(count+1:count+Param_fix.num_time_data(iexp));
    count = count + Param_fix.num_time_data(iexp);
    D_data = yhat(count+1:count+Param_fix.num_time_data(iexp)) + ...
        yhat_CI(count+1:count+Param_fix.num_time_data(iexp));
    figure(Afig); hold on;
    plot(time_vect,A_data,'-.');
    figure(Cfig); hold on;
    plot(time_vect,C_data,'-.');
    figure(Dfig); hold on;
    plot(time_vect,D_data,'-.');
    clear A_data C_data D_data;
    % update the master count vector
    count_master = count_master + 3*Param_fix.num_time_data(iexp);
end
% Then, add the labels to each graph.
figure(Afig);
xlabel('time');
ylabel('[A]');
title('Concentration of species A');
figure(Cfig);
xlabel('time');
ylabel('[C]');
title('Concentration of species C');
```

```
figure(Dfig);
xlabel('time');
ylabel('[D]');
title('Concentration of species D');
```