```
% lst_sqr_exmp.m ... examples of least-squares data fiting

clear


EXAMPLE = 4     % select which example to run 1, 2, 3, 4, 5, 6, 7, 8

if EXAMPLE == 1 % ... not polynomial, but linear in parameters

delta_t = 0.1;
noise = 0.2;
t_data = 0.1*[1:140]';                          % t_data ... column vector

a = [ 1.0 ; -0.5 ; 0.2 ; -0.1 ];                % 'true' values of parameters

X = [ sin(t_data) sin(2*t_data) sin(3*t_data) sin(4*t_data) ];  % design matrix
[points,n] = size(X);
y_true = X * a;

y_data = y_true  +  noise*randn(points,1);      % noise on y *only*

a_lls = inv(X'*X) * X'* y_data;                 % linear least-sqares parameters
y_lls = X * a_lls;
SE_data = sqrt(sum((y_data-y_lls).^2)/(points-n+1)) % std. err of data wrt model

Va = SE_data^2 * inv(X'*X);                     % parameter covariance
SE_fit = sqrt(diag(X * Va * X'));               % standard deviation of the fit

disp('     a           a_lls  +/-  da        (percent)       ');
disp([  a  a_lls   sqrt(diag(Va))       100*sqrt(diag(Va))./abs(a_lls) ]);

% SE_data =  0.18150
%      a           a_lls   +/-  da        (percent)
%     1.000000    0.960019    0.021834    2.274381
%    -0.500000   -0.518274    0.021721    4.190941
%     0.200000    0.229585    0.021828    9.507391
%    -0.100000   -0.071647    0.021676   30.253335


figure(1)
 clf
  plot( t_data, y_true,'-r', t_data,y_lls,'--b', t_data,y_data,'og', ...
       t_data, y_lls + 1.96*SE_fit,'.k', t_data, y_lls - 1.96*SE_fit,'.k' );
   title('Example 1'); xlabel('x');  ylabel('y');
   legend('true','fit','data','+/- 95% c.i.','',4)

end % EXAMPLE 1

%
```

```
if EXAMPLE == 2 % ... linear fit in multi-dimensions

delta_t = 0.01;
points = 100;
noise = 10;
t_data = delta_t*[1:points]';

a = [ 5 ; 2 ; 10 ; 100 ];                       % 'true' parameter values
x1_data = sin(2*pi*3*t_data);                   % like a displacement
x2_data = 2*pi*3*cos(2*pi*3*t_data);            % like a velocity
y_true = a(1)*tanh(x2_data) + a(2)*x2_data + a(3)*x1_data + a(4)*x1_data.^3;
y_data = y_true + noise*randn(points,1);        % add msmsnt noise

X = [ tanh(x2_data)  x2_data  x1_data x1_data.^3 ];     % design matrix
[points,n] = size(X);

a_lls = inv(X'*X) * X'* y_data;         % linear least-sqares parameters
y_lls = X * a_lls;
SE_data = sqrt(sum((y_data-y_lls).^2)/(points-n+1)) % std. err of data wrt model

Va = SE_data^2 * inv(X'*X);                     % parameter covariance
SE_fit = sqrt(diag(X * Va * X'));               % standard deviation of the fit

disp('     a           a_lls  +/-  da        (percent)       ');
disp([  a  a_lls   sqrt(diag(Va))       100*sqrt(diag(Va))./abs(a_lls) ]);

% SE_data =  11.424
%      a           a_lls  +/-  da        (percent)
%      5.00000     8.21918     2.88087    35.05054
%      2.00000     1.74656     0.21243    12.16247
%     10.00000     5.77753     5.10875    88.42454
%    100.00000   105.75584     6.46211     6.11041


figure(1)
clf
subplot(211)
  plot( t_data, y_true,'-r', t_data,y_lls,'--b', t_data,y_data,'og', ...
       t_data, y_lls + 1.96*SE_fit,'.k', t_data, y_lls - 1.96*SE_fit,'.k' );
   title('Example 2'); xlabel('t');  ylabel('y');
   legend('true','fit','data','+/- c.i. ','',3)
subplot(223)
 plot(x1_data,y_lls,'-r', x1_data, y_lls,'--b', x1_data,y_data,'og', ...
       x1_data, y_lls + 1.96*SE_fit,'.k', x1_data, y_lls - 1.96*SE_fit,'.k' );
   xlabel('x_1');  ylabel('y');
subplot(224)
 plot(x2_data,y_true,'-r', x2_data, y_lls,'--b', x2_data,y_data,'og', ...
       x2_data, y_lls + 1.96*SE_fit,'.k', x2_data, y_lls - 1.96*SE_fit,'.k' );
    xlabel('x_2');  ylabel('y');

end % EXAMPLE 2

%
```

```
if EXAMPLE == 3 % ... power-law fit
                % ... not linear in parameters, but transformable??

x_data = [2:2:100]' * 0.1;
points = length(x_data);
noise = 0.20;

a = [ 0.1 ; 2.0 ];                              % 'true' values of parameters
X = [ ones(points,1) log(x_data) ];             % design matrix
[points,n] = size(X);

y_true = a(1) * x_data .^ a(2);
y_data = y_true  +  noise*randn(points,1);
y_data = max(y_data,0.0001);    % can not take log of non-positive values

a_lls = inv(X'*X) * X'* log(y_data);            % linear least-sqares parameters

y_lls = exp(X*a_lls);                           % estimated fit

a_lls(1) = exp(a_lls(1));                       % actual a_1 parameter

SE_data= sqrt(sum((y_data-y_lls).^2)/(points-n+1)) % std. err of data wrt model

Va     = SE_data^2 * inv(X'*X);                 % parameter covariance
SE_fit = sqrt(diag(X * Va * X'));               % standard deviation of the fit

disp('     a           a_lls  +/-  da        (percent)       ');
disp([  a  a_lls   sqrt(diag(Va))       100*sqrt(diag(Va))./abs(a_lls) ]);

% SE_data =  0.55349
%      a           a_lls  +/-  da        (percent)
%    1.0000e-01   1.3374e-01   1.4407e-01   1.0772e+02
%    2.0000e+00   1.8073e+00   8.8927e-02   4.9205e+00

figure(1)
 clf
 subplot(211)
  plot( x_data, y_true,'-r', x_data,y_lls,'--b', x_data,y_data,'og', ...
       x_data, y_lls + 1.96*SE_fit,'.k', x_data, y_lls - 1.96*SE_fit,'.k' );
  title('Example 3(a) (log-transformed linear least squares)');
  ylabel('y');
  legend('true','fit','data','+/- 95% c.i.','',2)

% Levenberg-Marquardt ...
[a_lm, X2,SE_a,SE_fit,corr,R_sq,cvg_hst] = ...
                              lm ( 'fit3', a_lls, x_data, y_data );

y_lm = a_lm(1) * x_data .^ a_lm(2);             % estimated fit

disp('     a           a_lm   +/-  da        (percent)       ');
disp([  a  a_lm    SE_a   100*SE_a./a_lm ]);

%      a           a_lm   +/-  da        (percent)
%   0.1000000   0.0814411   0.0062507   7.6751474
%   2.0000000   2.0885937   0.0360671   1.7268598

figure(1)
 subplot(212)
  plot( x_data, y_true,'-r', x_data,y_lm,'--b', x_data,y_data,'og', ...
       x_data, y_lm + 1.96*SE_fit,'.k', x_data, y_lm - 1.96*SE_fit,'.k' );
  title('Example 3(b) (non-linear least squares)');
  xlabel('x');  ylabel('y');
 lm_plots( x_data, y_data, y_lm, SE_fit, cvg_hst, 'lstsqr_2' );

end % EXAMPLE 3
%
```

```
if EXAMPLE == 4 % ...  a non-linear least sqares problem with exponentials

  delta_x = 0.1;
  points = 200;
  noise = 0.10;                 % standard deviation of the noise
  x_data = [1:points]'*delta_x; % values of the independent variable

  a = [  4   2   1   10 ]';     % 'true' value of the parameters


  y_true = a(1)*exp(-x_data/a(2)) + a(3)*x_data.*exp(-x_data/a(4));
  y_data = y_true + noise*randn(points,1);

  a_init     = [ 3  1  2  7 ]'; % initial guess for parameters

  options = [0 0.01 0.01 ];

% Levenberg-Marquardt ...
  [a_lm,X2,SE_a,SE_fit,corr,R2,cvg_hst] = lm('fit4', a_init, x_data, y_data );
   y_lm = feval('fit4',x_data,a_lm);

  disp('     a           a_lm  +/-  da        (percent)       ');
  disp([  a  a_lm   SE_a   100*SE_a./a_lm ]);

%    a           a_lm   +/-  da        (percent)
%    4.000000    3.961656    0.055715    1.406368
%    2.000000    2.061105    0.060654    2.942775
%    1.000000    0.987223    0.011396    1.154318
%   10.000000   10.102611    0.085156    0.842907


  figure(1);
   clf
   plot(x_data,y_true,'-r', x_data, y_lm,'--b', x_data,y_data,'og', ...
      x_data, y_lm + 1.96*SE_fit,'.k', x_data, y_lm - 1.96*SE_fit,'.k' );
    title('Example 4'); xlabel('x');  ylabel('y');
    legend('true','fit','data','+/- 95% c.i.','',3)
   lm_plots( x_data, y_data, y_lm, SE_fit, cvg_hst, 'lstsqr_4' );

end % EXAMPLE 4

%
```

```
if EXAMPLE == 5 % ... rational complex polynomial
                % ... not linear in parameters, but transformable?

  i = sqrt(-1);

% for small delta_f and low noise, OLS works fine
%  for example : delta_f < 0.1, and noise < 5, OLS works fine ...
% delta_f = 0.1; noise =   1.0;                % ... good
  delta_f = 0.2; noise =  10.0;                % ... bad
  points = floor(5.0/delta_f);
  w_data = [-points:points]'*delta_f;   % values of the independent frequencies
  iw_data = i*w_data;
  points = 2*points + 1;

% 'true' parameter values ...
   a = [ 10  -2  1    0.1   0.3  0.015  0.015  ]';

% H is Hermetian symmetric ... frequency response function
  H_true = (a(1)*iw_data + a(2)*iw_data.^2 + a(3)*iw_data.^3 ) ./ ...
     (1 + a(4)*iw_data + a(5)*iw_data.^2 + a(6)*iw_data.^3 + a(7)*iw_data.^4);

  M = length(w_data);
  % noise must be Hermetian symmetric
  noise_r = randn((M-1)/2,1);           % real part of noise
  noise_i = randn((M-1)/2,1);           % imaginary part of noise
  H_data = H_true + noise*[ noise_r((M-1)/2:-1:1) ; 0 ;  noise_r ] + ...
                 i*noise*[ noise_i((M-1)/2:-1:1) ; 0 ; -noise_i ];

  % design matrix [X] includes measurements {H_data} ... !!
  X = [   iw_data,   iw_data.^2, iw_data.^3,...
-H_data.*iw_data, -H_data.*iw_data.^2, -H_data.*iw_data.^3, -H_data.*iw_data.^4 ];

  [points,n] = size(X);

   a_lls = inv(X'*X) * X' * H_data;     % linear least-sqares parameters
                                        % same eq'n, even though complex-valued!

   a_lls = real(a_lls);                 % imaginary part is just round-off error

   H_lls_err = X*a_lls;        % this produces errors, should be a red flag!
   H_lls = fit5 ( w_data, a_lls );

   SE_data = sqrt(sum( abs(H_data-H_lls).^2 ) / (points-n+1)) % std.err data

   Va = SE_data^2 * inv(X'*X);          % parameter covariance
   SE_lls = sqrt(diag(X * Va * X'));    % standard deviation of the fit

   disp('     a           a_lls  +/-  da        (percent)       ');
   disp(real([  a  a_lls   sqrt(diag(Va))       100*sqrt(diag(Va))./abs(a_lls) ]));

% SE_data =  53.256
%     a           a_lls  +/-  da        (percent)
%   1.0000e+01   6.4797e+00   7.6288e+00   1.1773e+02
%  -2.0000e+00  -1.2357e+00   9.4737e-01   7.6669e+01
%   1.0000e+00   7.2744e-01   5.4972e-01   7.5569e+01
%   1.0000e-01   6.1165e-02   1.0296e-01   1.6833e+02
%   3.0000e-01   2.6569e-01   4.5833e-02   1.7250e+01
%   1.5000e-02   9.2873e-03   8.9263e-03   9.6113e+01
%   1.5000e-02   1.2279e-02   3.0847e-03   2.5121e+01


%
```

```
  figure(1);
   clf
   subplot(211)
    plot(w_data,abs(H_true),'-r', w_data, abs(H_lls),'--b' , ...
         w_data,abs(H_lls_err),'--g', ...
         w_data,abs(H_data),'og', ...
         w_data, abs(H_lls) + 1.96*abs(SE_lls),'.k', ...
         w_data, abs(H_lls) - 1.96*abs(SE_lls),'.k' );
     axis([ -5 5 0 200 ])
     title('Example 5(a)');  ylabel('|H(f)|');


% Levenberg-Marquardt ...

   M = length(w_data);
   a = real(a);
   H_data_ir = [ imag(H_data(1:(M-1)/2+1)) ; real(H_data((M-1)/2+2:M)) ];

  a_init = a_lls;       % a good initial guess


%        prnt  MaxIter  eps1  eps2  eps3  eps4 lambda0  Lup Ldn  Update
   opts = [ 3,     100, 1e-5, 1e-6, 1e-5, 1e-3,   1e-1,  11,  9, 1 ];

   a_min = a_init - 10;  a_max = a_init + 10;
   dp = 1e-5; % finite-diff increment ... must be SMALL to work for this example
   weight = abs(H_data_ir.^1);  % weight larger values more.
   [a_lm,X2,SE_a,SE_fit,corr,R2,cvg_hst] = ...
   lm('fit5b',a_init,w_data,H_data_ir, weight,dp,a_min,a_max,0,opts);

   H_lm = fit5 ( w_data, a_lm );

   SE_fit = [ SE_fit(M:-1:(M-1)/2+2) ; 0 ; SE_fit((M-1)/2+2:M) ] +  ...
            [ SE_fit(1:(M-1)/2) ; SE_fit((M-1)/2+1:-1:1) ] * sqrt(-1);

   disp('     a           a_lm   +/-  da        (percent)       ');
   disp(real([  a  a_lm   SE_a   100*SE_a./a_lm ]));

%     a           a_lm   +/-  da        (percent)
%   1.0000e+01   1.0897e+01   6.7395e-01   6.1849e+00
%  -2.0000e+00  -2.1057e+00   1.1021e-01  -5.2338e+00
%   1.0000e+00   1.1203e+00   8.4389e-02   7.5325e+00
%   1.0000e-01   1.0415e-01   6.7515e-03   6.4825e+00
%   3.0000e-01   2.9701e-01   1.8584e-03   6.2568e-01
%   1.5000e-02   1.5686e-02   1.1460e-03   7.3062e+00
%   1.5000e-02   1.4740e-02   2.6720e-04   1.8128e+00


  figure(1);
   subplot(212)
    plot( w_data,abs(H_true),'-r',   w_data,  abs(H_lm),'--b', ...
%         w_data,real(H_lm),'--b',  w_data,imag(H_lm),'--b', ...
%         w_data,real(H_data),'og',  w_data,imag(H_data),'og', ...
         w_data,abs(H_data),'og', ...
         w_data, abs(H_lm) + 1.96*(SE_fit),'.k', ...
         w_data, abs(H_lm) - 1.96*(SE_fit),'.k' );
    title('Example 5(b)'); xlabel('f');  ylabel('|H(f)|');
    axis([ -5 5 0 200 ])
lm_plots(w_data,abs(H_data), abs(H_lm),abs(SE_fit), cvg_hst,'lstsqr_5' );

end % EXAMPLE 5

%
```

```
if EXAMPLE == 6 % ... Impuse Response Function

  i = sqrt(-1);

  delta_t = 0.10;
  points = 200;
  noise = 1.00;
  t_data = [1:points]'*delta_t; % values of the independent variable

  a = [ 6 ;  3 ; -0.1 ; 12 ; 9 ; -0.2 ];       % 'true' parameter values

  y_true = a(1)*exp((i*a(2)+a(3))*t_data) + a(4)*exp((i*a(5)+a(6))*t_data);
  y_true = real(y_true);
  y_data = y_true + noise*randn(points,1);

% Levenberg-Marquardt ...
   a_init = a .* (1 + 0.10*randn(6,1));         % random initial guess
   dp = 1e-6; % finite-diff increment ... must be SMALL to work for this example
   [a_lm,X2,SE_a,SE_fit,corr,R2,cvg_hst] = ...
               lm('fit6', a_init, t_data, y_data, 1, dp );

    y_lm = fit6 ( t_data, a_lm );

   disp('     a           a_lm  +/-  da        (percent)       ');
   disp(real([  a  a_lm   SE_a   100*SE_a./abs(a_lm) ]));

%    a            a_lm    +/-  da          (percent)
%   6.0000e+00   5.9522e+00   3.3776e-01   5.6745e+00
%   3.0000e+00   3.0054e+00   5.9066e-03   1.9653e-01
%  -1.0000e-01  -9.8110e-02   8.8750e-03   9.0460e+00
%   1.2000e+01   1.2029e+01   4.5709e-01   3.8000e+00
%   9.0000e+00   9.0246e+00   7.1360e-03   7.9073e-02
%  -2.0000e-01  -1.9498e-01   1.0350e-02   5.3082e+00

  figure(1)
   clf
   plot(t_data,y_true,'-r', t_data, y_lm,'--b', t_data,y_data,'og', ...
         t_data, y_lm + 1.96*SE_fit,'.k', t_data, y_lm - 1.96*SE_fit,'.k' )
     title('Example 6'); xlabel('t'); ylabel('y: true, fit, and data');
     legend('true','fit','data','+/- 95% c.i.','',1)
 lm_plots( t_data, y_data, y_lm, SE_fit, cvg_hst, 'lstsqr_6' );

end % EXAMPLE 6

%
```

```
if EXAMPLE == 7 % ... Auto-regressive, Moving-Average (ARMA) model ID

 global nA  nB  u_data

 delta_t = 0.01;
 points = 200;
 noise = 0.5;
 t_data = [1:points]'*delta_t;  % values of the independent variable

 Ca = [ 4 -3 2 -1 ]/4; Cb = [ 2 4 6 8 ]/4;      % ARMA coefficients

 u_true = randn(points,1);
 y_true= filter(Cb,Ca,u_true);
 y_data = y_true + noise*randn(points,1);
 u_data = u_true + noise*randn(points,1);


 nA = length(Ca);
 nB = length(Cb);

% linear least squares solution
 for n = nA+nB+1 : points-nA-nB;
     yn(n-nA-nB) = y_data(n);
     % the design matrix [X] includes measurements {y_data} ... !!
     X(n-nA-nB,:) = [ -y_data(n-1:-1:n-nA+1)' u_data(n-0:-1:n-nB+1)' ];
 end
 a_lls = X \ yn(:);

 a_lls = [1 ; a_lls];

 Ca_lls = a_lls(1:nA);
 Cb_lls = a_lls(nA+1:nA+nB);

 y_fit = filter(Cb_lls, Ca_lls, u_data);

 SE_lls = sqrt(sum( abs(y_true-y_fit).^2 ) / (points-n+1))

 Va = SE_lls^2 * inv(X'*X);                     % parameter covariance
 SE_fit = sqrt(diag(X * Va * X'));              % standard deviation of the fit
 SE_fit = [ zeros(nA+nB,1) ; SE_fit ; zeros(nA+nB,1) ];

 disp(' a     a_fit    +/-     da          (percent)       ');
 disp([ [Ca';Cb']   [Ca_lls;Cb_lls]    [0;sqrt(diag(Va))]   [0;100*sqrt(diag(Va))].
/abs(a_lls) ]);

%      a           a_fit  +/-  da        (percent)
%    1.00000     1.00000     0.00000     0.00000
%   -0.75000    -0.51620     0.16775    32.49682
%    0.50000     0.20277     0.20060    98.92857
%   -0.25000    -0.10257     0.13518   131.78608
%    0.50000     0.48078     0.34854    72.49520
%    1.00000     1.05480     0.35355    33.51829
%    1.50000     1.72457     0.38728    22.45682
%    2.00000     2.14714     0.43508    20.26309

% total least squares
Y = zeros(points - nA  , nA);
U = zeros(points - nB  , nB);
for k = 1:points-nA-1
   Y(k,:) = y_data( k+nA-1 : -1 : k );
end
for k = 1:points-nB-1
   U(k,:) = u_data( k+nB-1 : -1 : k );
end

[UU,SS,VV] = svd([Y , -U]);
```

```
figure(334); semilogy(diag(SS),'o')

a_tls = VV(:,nA+nB) / VV(1,nA+nB);

 Ca_tls = a_tls(1:nA);
 Cb_tls = a_tls(nA+1:nA+nB);

 y_fit = filter(Cb_tls, Ca_tls, u_data);

 SE_tls = sqrt(sum( abs(y_true-y_fit).^2 ) / (points-n+1))

figure(333)
   plot(t_data, y_data,'og', t_data,y_true,'-b', t_data,y_fit,'--r' )

%
```

```
 figure(1);
  clf
  subplot(211)
   plot( t_data,y_fit + 1.96*SE_fit,'.k', t_data,y_fit-1.96*SE_fit,'.k',  ...
         t_data,y_data,'og', t_data,y_true,'-b', t_data,y_fit,'--r' );
    title('Example 7(a)');
    ylabel('y');

% Levenberg-Marquardt ... use linear least squares solution as initial guess
  dp = 1e-2;
  [a_lm,X2,SE_a,SE_fit,corr,R2,cvg_hst] = lm('fit7', a_lls(2:nA+nB).*(1+0.1*randn(7
,1)), t_data, y_data,1,dp );

 a_lm = [ 1 ; a_lm ];
 Ca_lm = [a_lm(1:nA)];
 Cb_lm = a_lm(nA+1:nA+nB);

 y_lm = filter(Cb_lm, Ca_lm, u_data);

 SE_lm = sqrt(sum( abs(y_true-y_lm).^2 ) / (points-n+1))

 disp('    a             a_lls        a_tls        a_lm         +/- da     (percent
)');
 disp([ [Ca';Cb'] a_lls  a_tls    a_lm    [0;SE_a]    100*[0;SE_a]./abs(a_lm) ]);

%    a             a_lls         a_lm         +/- da     (percent)
%   1.0000e+00   1.0000e+00   1.0000e+00   3.9900e+02   3.9900e+04
%  -7.5000e-01  -5.2045e-01  -7.6628e-01   3.0580e+02   3.9907e+04
%   5.0000e-01   2.2590e-01   5.5311e-01   2.2059e+02   3.9881e+04
%  -2.5000e-01  -1.3508e-01  -2.9112e-01   1.1604e+02   3.9862e+04
%   5.0000e-01   4.9687e-01   6.0055e-01   2.3975e+02   3.9922e+04
%   1.0000e+00   1.1312e+00   1.0264e+00   4.0908e+02   3.9857e+04
%   1.5000e+00   1.6733e+00   1.4215e+00   5.6773e+02   3.9938e+04
%   2.0000e+00   2.1807e+00   2.1330e+00   8.5021e+02   3.9860e+04

 figure(1);
  subplot(212)
   plot( t_data, y_lm + 1.96*SE_fit,'.k', t_data, y_lm - 1.96*SE_fit,'.k', ...
         t_data, y_data,'og', t_data,y_true,'-b', t_data,y_lm,'--r');
    title('Example 7(b)')
    ylabel('y');
    xlabel('t');
 lm_plots( t_data, y_data, y_lm, SE_fit, cvg_hst, 'lstsqr_7' );

end % EXAMPLE 7

%
```

```
if EXAMPLE == 8 % ... Nonlinear least squares for linear dynamic system

 global u_data

 delta_t = 0.02;
 points = 1000;
 t_data = [1:points]'*delta_t;  % values of the independent variable
 u_true = randn(points,1)/sqrt(delta_t);        % unit white noise

 %          k1   k2   k3   c1    c2    c3
 a_true = [ 20 ; 50 ; 50 ; 1.0 ; 0.5 ; 0.1 ];
 u_data = u_true;                               % u_true for y_true
 y_true = fit8(t_data,a_true);

 noise_u = 1.0;                                 % noise on input msmnt
 noise_y = 0.2;                                 % noise on output msmnt

 u_data = u_true + noise_u*randn(points,1);     % noise on input msmnt
 y_data = y_true + noise_y*randn(points,1);     % noise on output msmnt

 figure(1)
  clf
   subplot(211)
    plot(t_data,u_true,t_data,u_data);
     ylabel('input, u')
   subplot(212)
    plot(t_data,y_true,t_data,y_data);
     ylabel('output, y')

 a_init = a_true.*(1+0.50*randn(6,1));
 a_min = [   1   1   1 0.001 0.001 0.001 ];
 a_max = [ 100 100 100 2.0   2.0   2.0 ];

 [a_lm,X2,SE_a,SE_fit,corr,R2,cvg_hst] = ...
                        lm('fit8', a_init, t_data, y_data, 1, 0.05, a_min, a_max);

 y_lm = fit8(t_data,a_lm);

 disp('    a_true         a_lm        +/- da     (percent)');
 disp([    a_true         a_lm         SE_a    SE_a ./ abs(a_lm) ]);

%    a_true         a_lm        +/- da     (percent)
%   2.0000e+01   1.9617e+01   7.3721e-02   3.7580e-03
%   5.0000e+01   5.2391e+01   1.1698e+00   2.2328e-02
%   5.0000e+01   4.9934e+01   6.3300e-01   1.2677e-02
%   1.0000e+00   8.1479e-01   9.0425e-03   1.1098e-02
%   5.0000e-01   2.4827e-01   9.9856e-02   4.0220e-01
%   1.0000e-01   3.9191e-01   6.3095e-02   1.6099e-01

 figure(2)
  clf
    plot(t_data,y_true,'-r', t_data,y_lm,'--b', t_data,y_data,'og', ...
         t_data, y_lm + 1.96*SE_fit,'.k', t_data, y_lm - 1.96*SE_fit,'.k' )
     legend('true','fit','data','+/- 95% c.i.','',1)
     ylabel('y'); xlabel('t')
 lm_plots( t_data, y_data, y_lm, SE_fit, cvg_hst, 'lstsqr_8' );

end % EXAMPLE 8
```