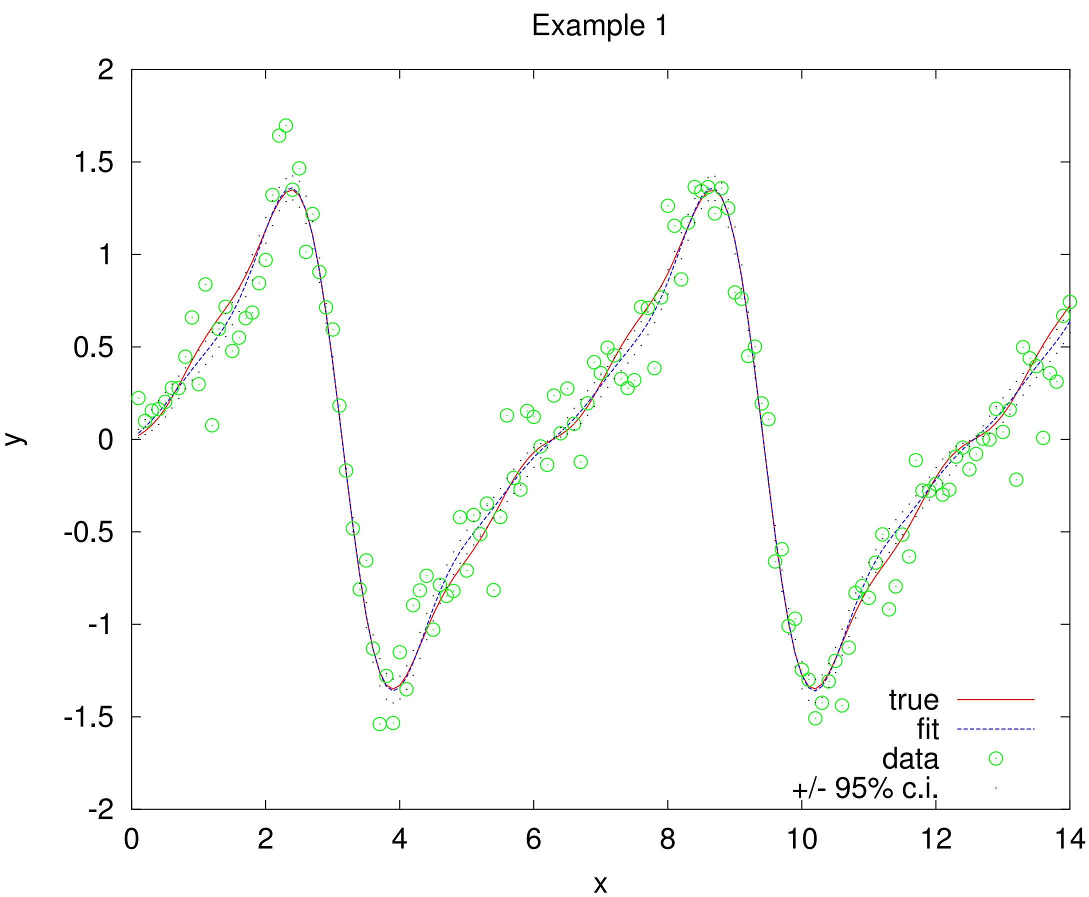
Eight Examples of Linear and Nonlinear Least Squares

CEE 699.04, ME 599.04 — System Identification — Fall, 2013
©Henri P. Gavin, September 25, 2015

1 Not polynomial, but linear in parameters.

$$\hat{y}(t_i; a) = a_1 \sin(t_i) + a_2 \sin(2t_i) + a_3 \sin(3t_i) + a_4 \sin(4t_i) \tag{1}$$

```
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
```



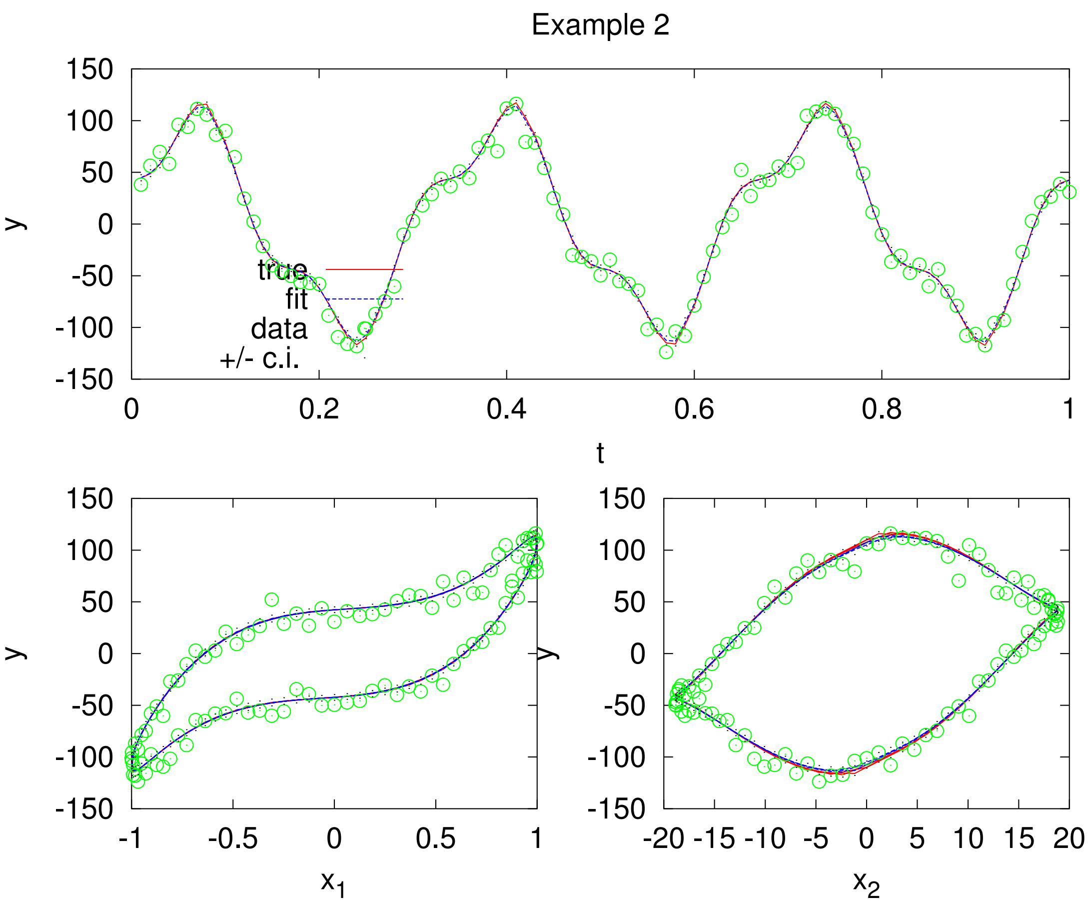
2 Linear Fit in Multi-Dimensions.

$$\hat{y}(x_1(t_i), x_2(t_i); a) = a_1 \tanh(x_1(t_i)) + a_2 x_2(t_i) + a_3 x_1(t_i) + a_4 x_1^3(t_i) \tag{2}$$

$x_1(t_i)$ is like a time-varying displacement

$x_2(t_i)$ is like a time-varying velocity

```
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
```



3 Power-Law Fit — NOT linear in parameters, but transformable?

$$\hat{y}(x_i; a) = a_1 x_i^{a_2} \tag{3}$$

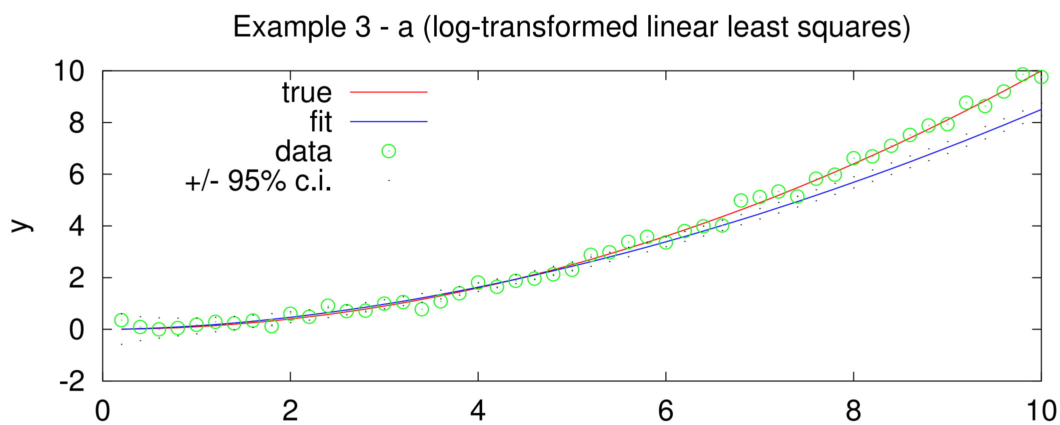
$$\log(\hat{y}(x_i; a)) = \log(a_1) + a_2 \log(x_i) \quad ?? \tag{4}$$

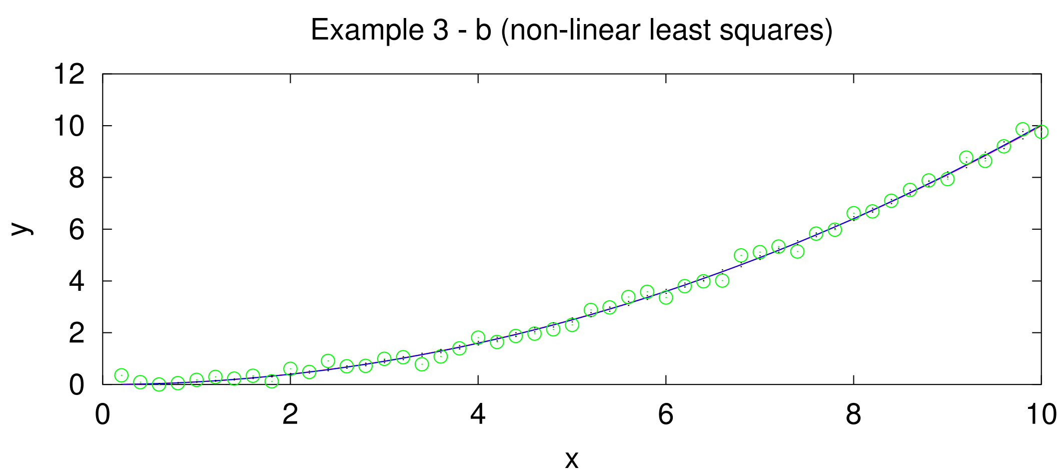
Parameters from Linear Least Squares applied to transformed equations: See Fig 3-a

```
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
```

Parameters from Nonlinear Least Squares: See Fig 3-b

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






```
% fit3.m - nonlinear least squares with power-law
function y_fit = fit3(x_data,a);
 y_fit = a(1) * x_data .^ a(2);
```

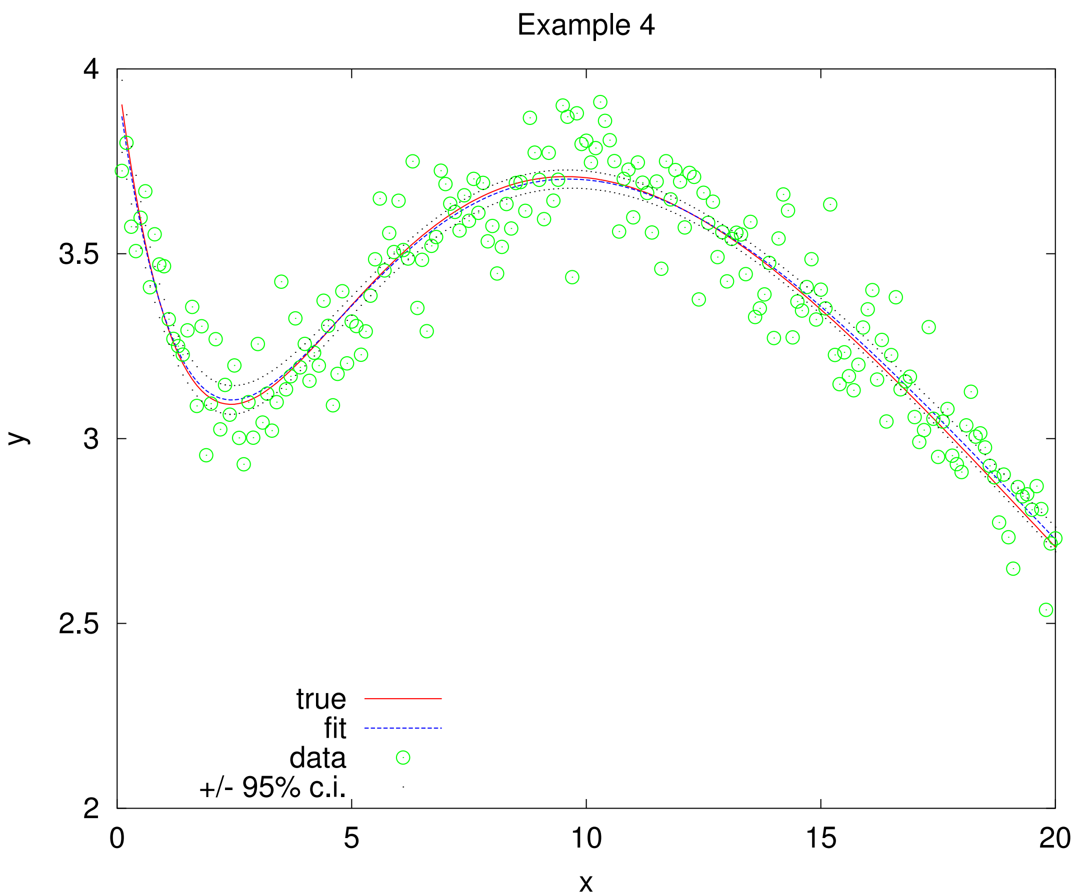
The log-error function under-values the data with larger measurement error.

4 Nonlinear Least Squares with Exponentials

$$\hat{y}(x_i; a) = a_1 \exp[-x_i/a_2] + a_3 \exp[-x_i/a_4] \tag{5}$$

Parameters from Nonlinear Least Squares:

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




```
% fit4.m - nonlinear least squares with exponentials
function y_fit = fit4(x_data,a)

 y_fit = a(1)*exp(-x_data/a(2)) + a(3)*x_data.*exp(-x_data/a(4));
```

5 Ratio of complex-valued polynomials — NOT linear in parameters, but transformable?

$$\hat{H}(\omega_k; a) = \frac{a_1(i\omega_k) + a_2(i\omega_k)^2 + a_3(i\omega_k)^3}{1 + a_4(i\omega_k) + a_5(i\omega_k)^2 + a_6(i\omega_k)^3 + a_7(i\omega_k)^4} \tag{6}$$

$$e_k = H(\omega_k)\left(1 + a_4(i\omega_k) + a_5(i\omega_k)^2 + a_6(i\omega_k)^3 + a_7(i\omega_k)^4\right) - \left(a_1(i\omega_k) + a_2(i\omega_k)^2 + a_3(i\omega_k)^3\right) \quad ?? \tag{7}$$

Parameters from Linear Least Squares applied to transformed equations: See Fig 5-a

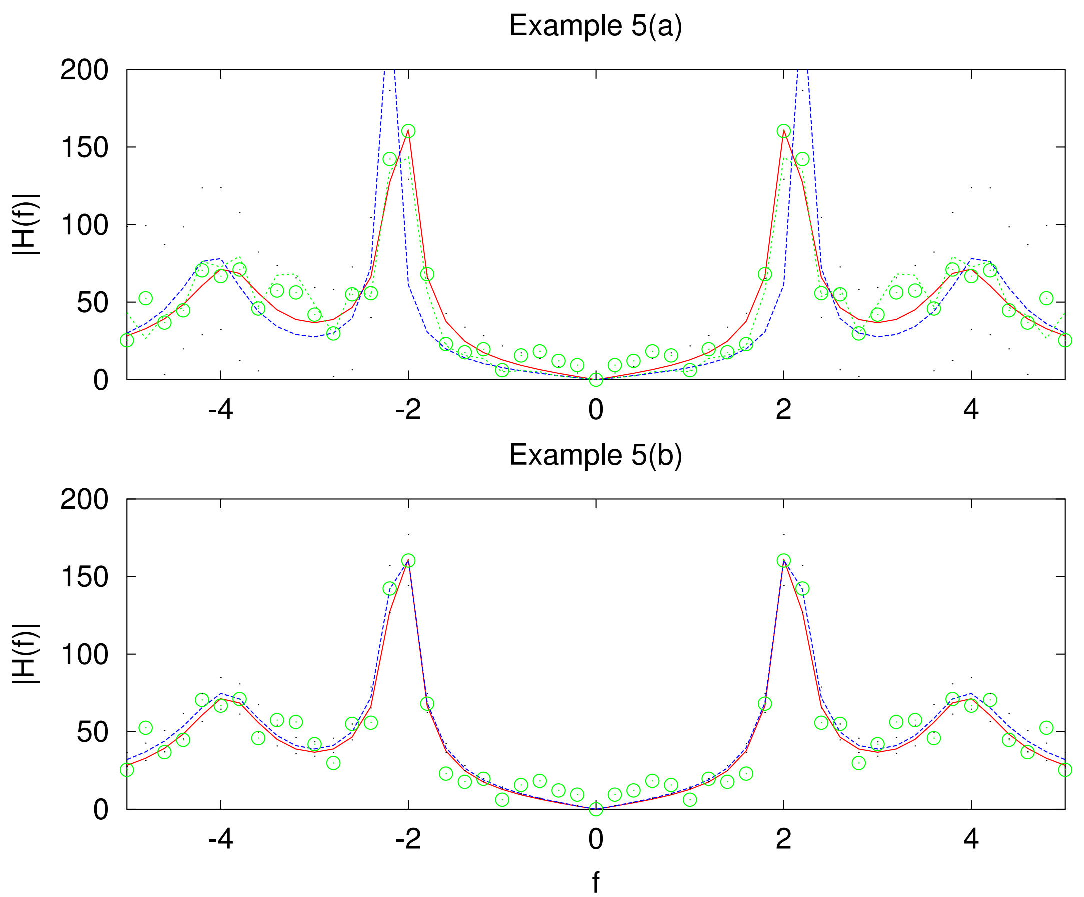
```
SE_data =  53.256
    a           a_lls  +/-  da         (percent)
  1.0000e+01   6.4797e+00   7.6288e+00   1.1773e+02
 -2.0000e+00  -1.2357e+00   9.4737e-01   7.6669e+01
  1.0000e+00   7.2744e-01   5.4972e-01   7.5569e+01
  1.0000e-01   6.1165e-02   1.0296e-01   1.6833e+02
  3.0000e-01   2.6569e-01   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
```

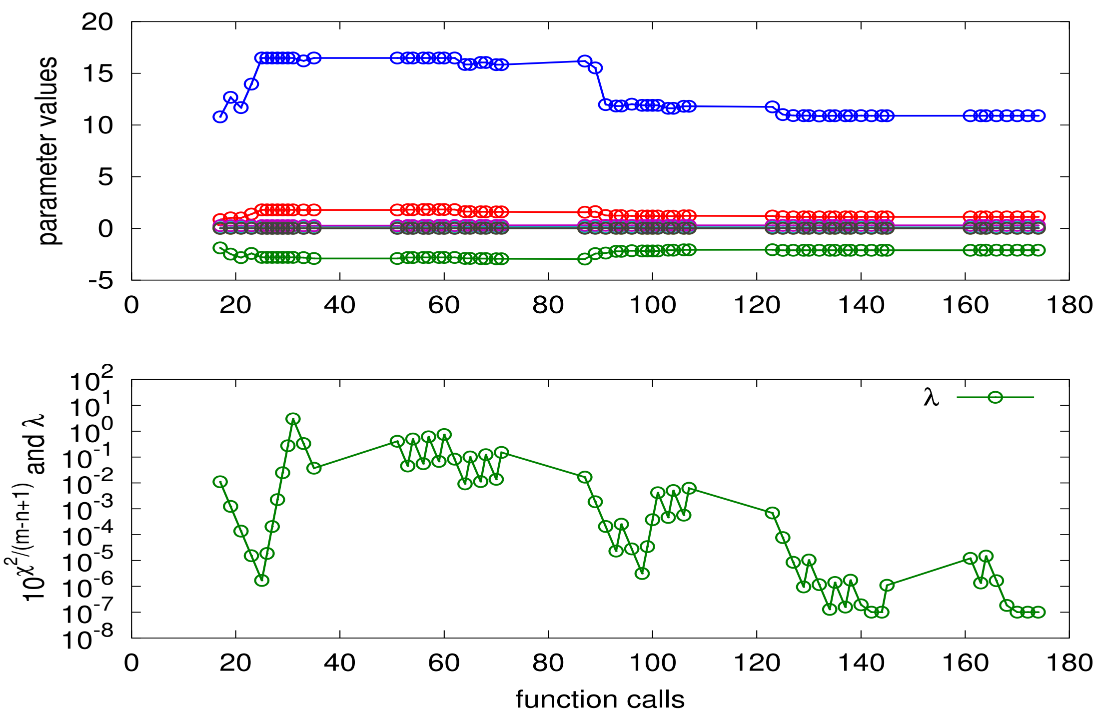
Parameters from Nonlinear Least Squares: See Fig 5-b

```
    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
```

```
% fit5.m - nonlinear least squares with ratio of complex-valued polynomials
function H_fit = fit5(w_data,a);

 iw_data = sqrt(-1) * w_data;

  H_fit = (a(1)*iw_data + a(2)*iw_data.^2  + a(3)*iw_data.^3 ) ./ ...
(1.0 + a(4)*iw_data + a(5)*iw_data.^2 + a(6)*iw_data.^3 + a(7)*iw_data.^4 );
```

```
% fit5b.m - ratio of complex-valued exponentials ...
% for use with Levenberg-Marquard fitting ... lm.m
% negative frequency has the imaginary part, positive frequency has the real part
function H_fit = fit5b(w_data,a);

  iw_data = sqrt(-1) * w_data;

  H_fit = (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);
  H_fit = [imag(y_fit(1:(M-1)/2+1)) ; real(y_fit((M-1)/2+2:M)) ];
```


Example 5(a)
|H(f)|
Example 5(b)
|H(f)|
f

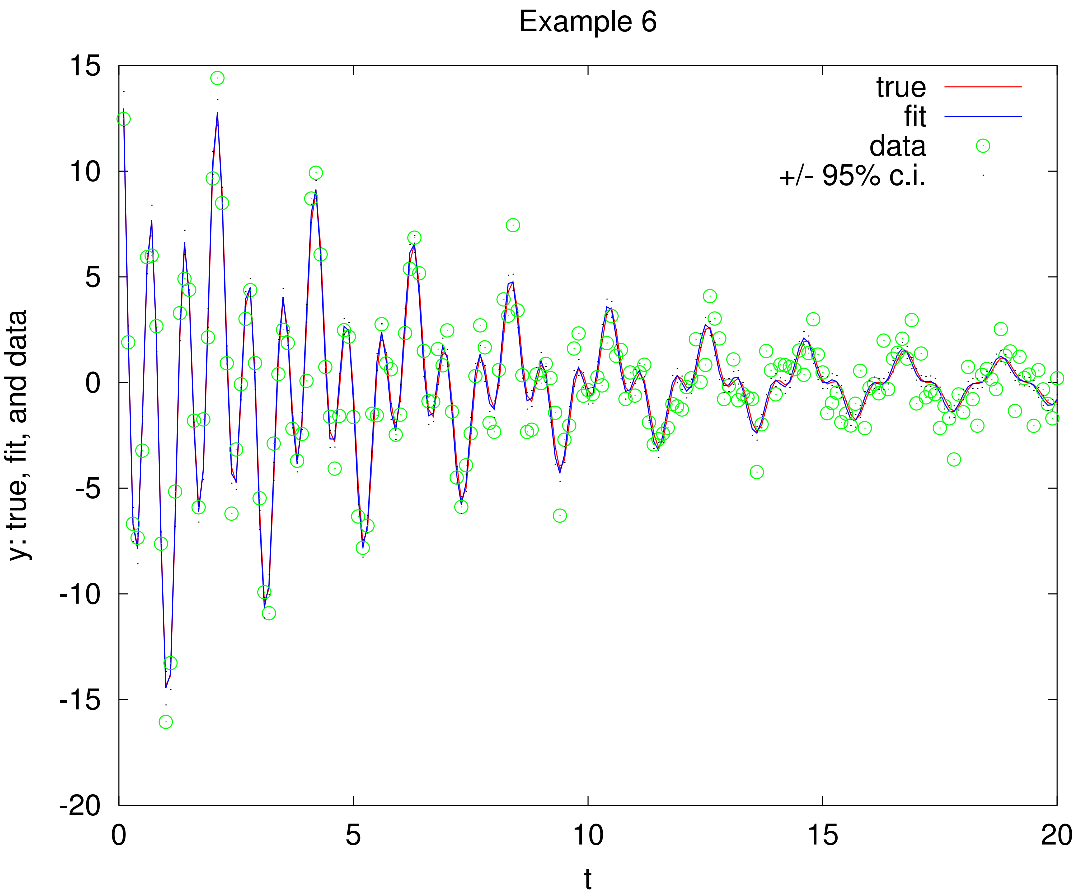

parameter values
10^{χ²/(m-n+1)} and λ
λ
function calls

6 Impulse Reponse Function

$$\hat{y}(t_k; a) = a_1 \exp[(ia_2 + a_3)t_k] + a_4 \exp[(ia_5 + a_6)t_k] \tag{8}$$

Parameters from Nonlinear Least Squares:

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




```
% fit6.m - impulse response function
function y_fit = fit6(t_data,a);

 i = sqrt(-1);

 y_fit = a(1)*exp((i*a(2)+a(3))*t_data) + a(4)*exp((i*a(5)+a(6))*t_data);

 y_fit = real(y_fit);  % taking the real part is like adding the complex conjugate
```

7 Auto Regressive - Moving Average Model — NOT linear in the parameters, but transformable?

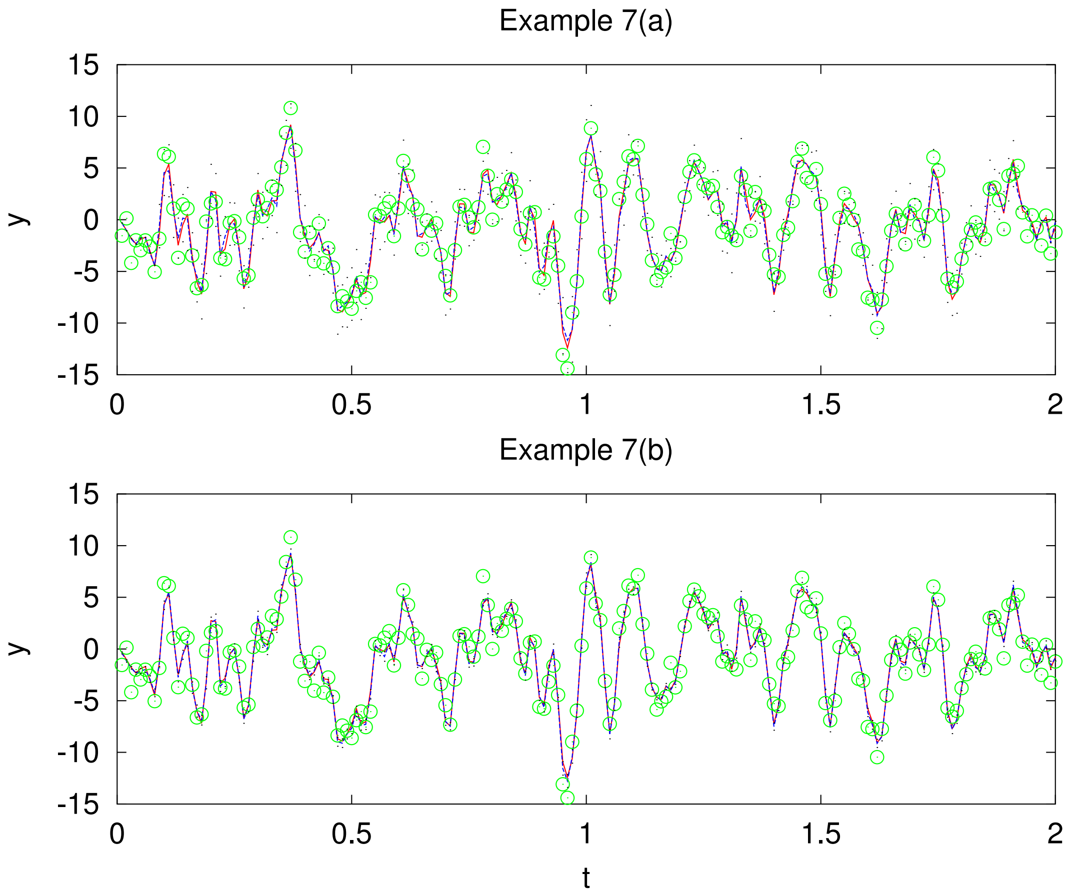
$$\hat{y}(t_i; a, b) = \sum_{k=1}^{3} a_k \ \hat{y}(t_{i-k}; a, b) + \sum_{k=0}^{3} b_k \ u(t_{i-k}) \tag{9}$$

$$\hat{y}(t_i; a, b) = \sum_{k=1}^{3} a_k \ y(t_{i-k}) + \sum_{k=0}^{3} b_k \ u(t_{i-k}) \quad ?? \tag{10}$$

Parameters from Linear Least Squares (Fig 7a) and Nonlinear Least Squares: (Fig 7b)

```
  a             a_lls         a_nls    +/- 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
  b             b_lls         b_nls    +/- db           (percent)
  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
```

```
% fit7.m - Auto-regressive moving-average (ARMA) model
function y_fit = fit7(x_data,Cab);

 global u_data  nA  nB

 Ca_fit = Cab(1:nA);
 Cb_fit = Cab(nA+1:nA+nB);

 y_fit = filter(Cb_fit, Ca_fit, u_data);
```


Example 7(a)
15
10
5
0
-5
-10
-15
y
0
0.5
1
1.5
2
Example 7(b)
15
10
5
0
-5
-10
-15
y
0
0.5
1
1.5
2
t

8 Identify Stiffness and Damping parameters directly— NOT linear in the parameters.

$$M\ddot{r}(t) + C\dot{r}(t) + Kr(t) = -Mhu(t) \tag{11}$$

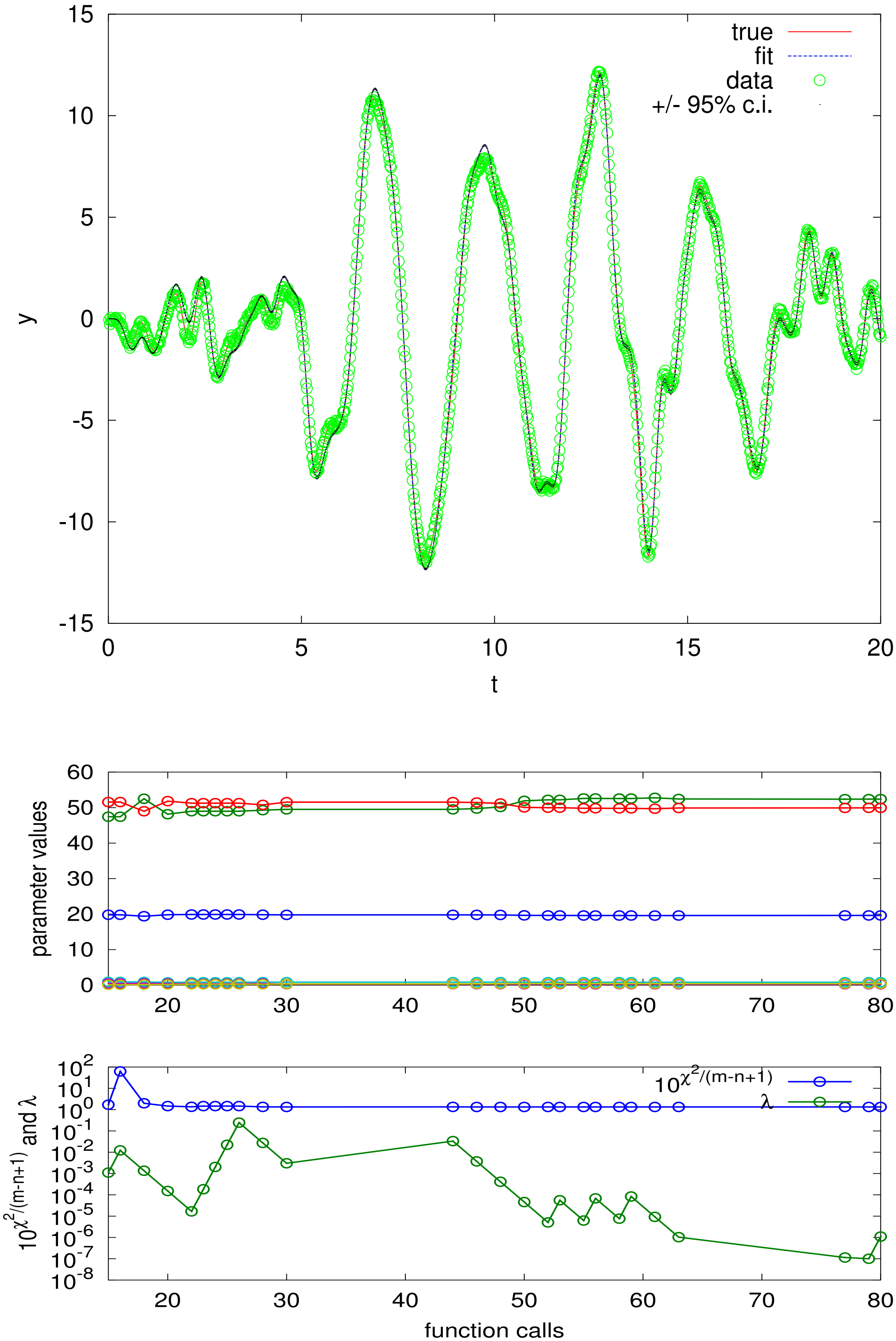
$$M = I_3 \quad C = \begin{bmatrix} c_1+c_2 & -c_2 & 0 \\ -c_2 & c_2+c_3 & -c_3 \\ 0 & -c_3 & c_3 \end{bmatrix} \quad K = \begin{bmatrix} k_1+k_2 & -k_2 & 0 \\ -k_2 & k_2+k_3 & -k_3 \\ 0 & -k_3 & k_3 \end{bmatrix} \quad h = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \tag{12}$$

$$\frac{d}{dt}\begin{bmatrix} r \\ \dot{r} \end{bmatrix} = \begin{bmatrix} 0 & I \\ -M^{-1}K & -M^{-1}C \end{bmatrix}\begin{bmatrix} r \\ \dot{r} \end{bmatrix} + \begin{bmatrix} 0 \\ -h \end{bmatrix} u(t) \tag{13}$$

$$y = \ddot{r}_3(t) + u(t) \tag{14}$$

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

```
% fit8.m - nonlinear least squares for a linear dynamic system
function y_fit = fit8(t_data,a);

 global u_data;

 k1 = a(1);              % stiffness parameter 1
 k2 = a(2);              % stiffness parameter 2
 k3 = a(3);              % stiffness parameter 3
 c1 = a(4);              % damping parameter 1
 c2 = a(5);              % damping parameter 2
 c3 = a(6);              % damping parameter 3

 Ks = [ k1+k2 -k2     0  ;       % stiffness matrix
        -k2    k2+k3 -k3 ;
         0    -k3     k3 ];

 Cs = [ c1+c2 -c2     0  ;       % damping matrix
        -c2    c2+c3 -c3 ;
         0    -c3     c3 ];

 Ms = eye(3);                    % mass matrix

 A = [ zeros(3) eye(3)  ;        % dynamic matrix
       -Ms\Ks   -Ms\Cs  ];

 B = [ 0; 0; 0; -1; -1; -1 ];    % input is acceleration of base

%C = [ 0  0  1   0   0   0 ];    % output msmnt is displ of DOF # 3
%C = [ 0  0  0   0   1   0 ];    % output msmnt is veloc of DOF # 2
 C = A(6,:);                     % output msmnt is accel of DOF # 3

 D = 0;

% damp(A)                        % check eigenvalues of dynamics matrix

 y_fit = lsim(A,B,C,D,u_data',t_data)';
```


true
fit
data
+/- 95% c.i.
y
t
parameter values
$10^{\chi^2/(m-n+1)}$ and λ
$10^{\chi^2/(m-n+1)}$
λ
function calls