Lab 7
Examples of General Linear and Non-linear Regression Problems

Handout – print version of Lecture on *Marine Modelling* February 25, 2019

Katja Fennel, Oceanography, Dalhousie University

7.1

1 Objectives

Objectives

We will work through two examples:

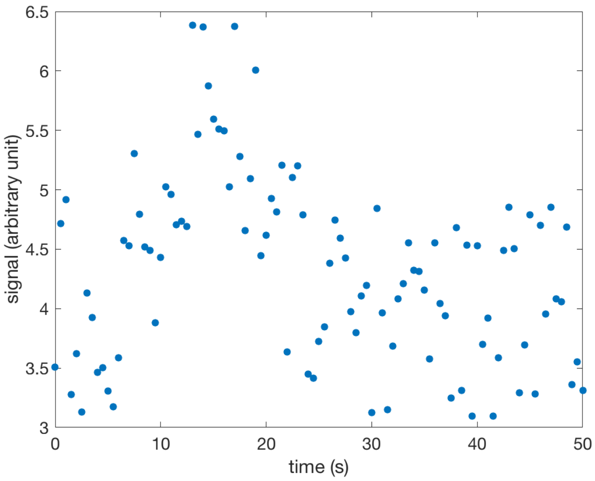
- a Gaussian peak superimposed on a constant background (general linear regression problem)
- an exponentially growing phytoplankton population (non-linear regression problem)

7.2

2 Example 1: "Defined" Gaussian Peak over constant Background

Example 1: "Defined" Gaussian Peak over constant Background

Suppose you have a data set that describes a background signal (with random noise) but that superimposed on the background is a contaminating signal that introduced a Gaussian peak at time=15s and with a standard deviation of 6s.



7.3

Example 1: "Defined" Gaussian Peak over constant Background

Suppose you are interested in the value of the background without the Gaussian contamination. The model function would look like this:

$$y(t) = a_1 + a_2 \ \exp(-\frac{(t-15)^2}{6^2})$$

You could determine this by general linear regression using the design matrix. Follow along with script `defined_Gaussian_peak.m`

First load data (t,y).

```
 load defined_Gaussian_peak_on_const_bg
```

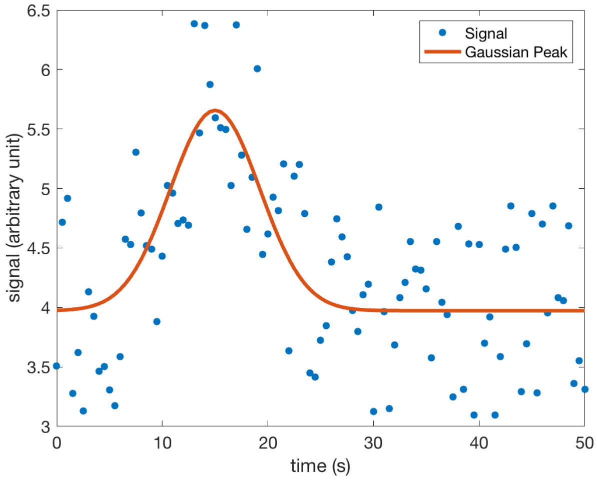
7.4

```
% build design matrix
A = [ones(length(t),1)  exp(-((t-15)/6).^2)'];

[U,S,V] = svd(A,0);    % SVD of the design matrix
W = diag(1./diag(S)); % Note: we didn't check
% for zero singular values
a = V*W*U'*y;          % our coefficients
[n m] = size(A);
redchisqu = sum((A*a-y).^2)/(n-m); % reduced
% Chi-squared
errmat = redchisqu*V*W.^2*V'; % covariance matrix
```

7.5

Results:

```
 a =  3.9728   1.6842
 sa = sqrt(diag(errmat))
 sa =  0.0668   0.1732
```



7.6

3 Example 2: Exponential Phytoplankton Growth

Example 2: Exponential Phytoplankton Growth

A simple non-linear problem would be exponential phytoplankton growth.

When phytoplankton is suddenly provided with sufficient light and nutrients it will start to grow nearly exponentially for a while.

$$\frac{\partial P}{\partial t} = \mu P$$

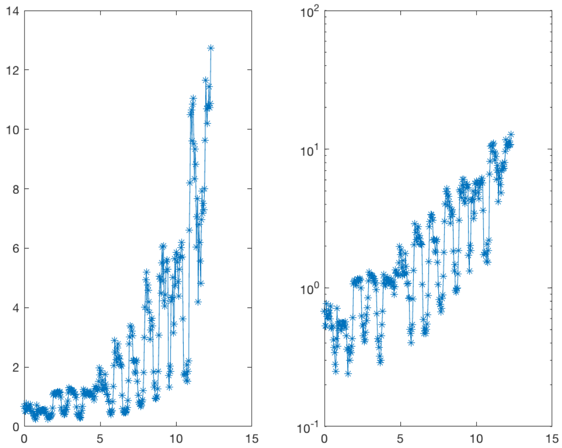
Here, μ is the specific growth rate with units of inverse time. Integration yields

$$P = P_0 \exp(\mu t)$$

or, we may assume there is a background population that doesn’t participate in the growth. Then

$$P = P_1 + P_2 \exp(\mu t)$$

Consider the Lobo data from spring 2008 (x is time; y is chlorophyll concentration): 7.7



We want to fit our above model to the data (we may be especially interested in the growth parameter μ). 7.8

Model:

$$P = P_1 + P_2 \exp(\mu t)$$

Since our model is non-linear we are going to use the non-linear least squares fitting routine `nlleasqr.m`

The routine is called thus:

```
[f,a,kvg,iter,corp,covp,covr,stdresid,Z,r2] = ...
                  nlleasqr(x,y,ain,’func_name’);
```

with input parameters `x, y` representing our data, `ain` is a vector that holds the initial guess of our parameters, and `func_name` is the name of a Matlab function that contains our model (we need to provide this function).

Follow along using script `non_lin_phyto_fit.m` (you’ll also need `modfunc1.m` and `dfdp.m`).

First a quick look at `modfunc1.m`: 7.9

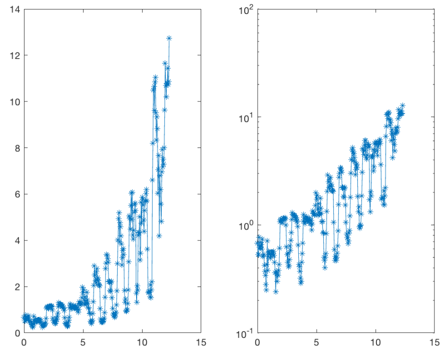
```
function out = modfunc1(x,a)
out = a(1) + a(2)*exp(a(3)*x);
```

7.10

```
% load data
load chl_spring08
% y contains chlorophyll fluorescence [micro g/l]
% x contains time in days since 2001-1-1 [UTC]

% QC: clean up NaNs in y
bad = find(isnan(y));
y(bad) = [];
```

```
x(bad) = [];

% modify x so that it’s relative to beginning of
% the data set
x = x-x(1);

% plot  (we need to get an idea of values for
% initial parameter guess)
figure
subplot(1,2,1)
plot(x,y,’*-’)
subplot(1,2,2)
semilogy(x,y,’*-’)
```

7.11



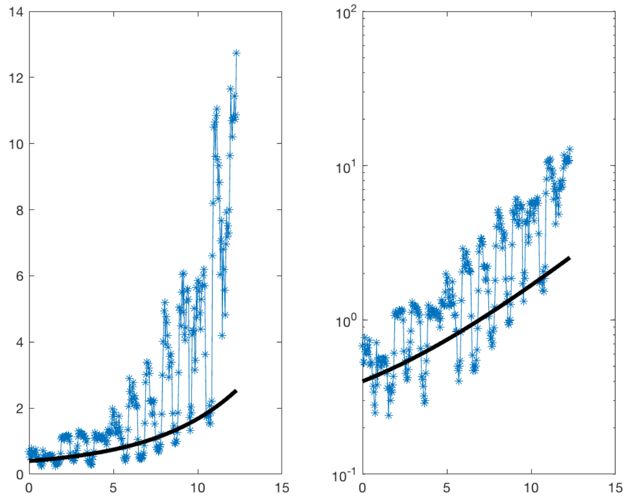
We need first guess values for all three parameters.

```
% first guess: baseline 0.2, ini pop 0.2,
% specific growth rate 0.2  (doubling
% in five days)
ain = [0.2 0.2 0.2];
```

7.12

```
% plot initial guess model on top of your data
figure
subplot(1,2,1)
plot(x,y,’*-’)
hold on
plot(x,modfunc1(x,ain),’k’,’LineWidth’,3)
%
subplot(1,2,2)
semilogy(x,y,’*-’)
hold on
semilogy(x,modfunc1(x,ain),’k’,’LineWidth’,3)
```

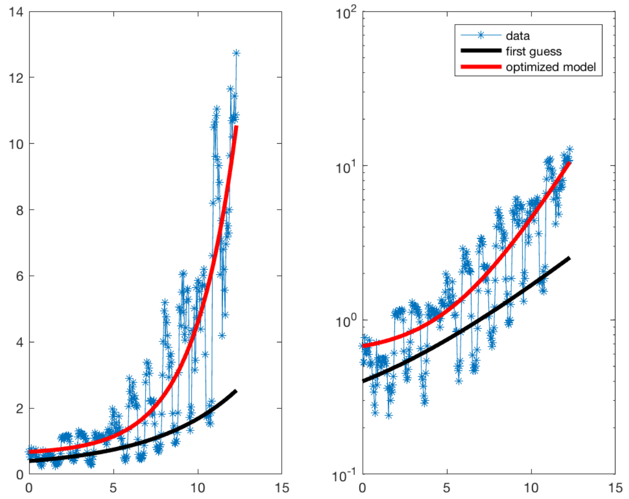
7.13



7.14

```
% NOW FIT to model function
% a1 + a2*exp(a3*t)
[f,a,kvg,iter,corp,covp,covr,stdresid,Z,r2] = ...
            nlleasqr(x,y,ain,'modfunc1');
% a is your optimal parameter vector
% let's see how much better it a does:
figure
subplot(1,2,1)
plot(x,y,'*-')
hold on
plot(x,modfunc1(x,ain),'k','LineWidth',3)
plot(x,modfunc1(x,a),'r','LineWidth',3)
%
subplot(1,2,2)
semilogy(x,y,'*-')
hold on
semilogy(x,modfunc1(x,ain),'k','LineWidth',3)
semilogy(x,modfunc1(x,a),'r','LineWidth',3)
legend('data','first guess','optimized model')
```

7.15



7.16

```
% kvg: flag to say whether convergence was
%      achieved before the routine gave up
% iter: number of iterations that was used
% covp: covariance matrix of the coefficients
%       (square roots of diagonal elements are
%       uncertainties (sigmas) of coefficients)
% r2: overall correlation coefficient

coefficients = a'
sigmas = sqrt(diag(covp))'
R2 = r2

% coefficients = 0.6025    0.0733    0.3995
% sigmas       = 0.1506    0.0255    0.0293
% R2           = 0.7862
```

7.17

Given:

```
% coefficients = 0.6025    0.0733    0.3995
% sigmas       = 0.1506    0.0255    0.0293
% R2           = 0.7862
```

$\mu = 0.40 \pm 0.03$ per day

78.6% of the variability in the data is explained by our model.

What about P1 and P2? Do we really need P1?

$$P = P_1 + P_2 \exp(\mu t)$$

Redo the fit with modified function without P1 – How would our function look like?

$$P = P_0 \exp(\mu t)$$

7.18