Problem (text 13-29): The table below gives the population of a small but growing suburb over a twenty year period.

Year	0	5	10	15	20
Population	100	200	450	950	2000

The growth is assumed to be exponential:

population = α*exp(βt), where t is a time in years

What values of α and β best fit the data?

What can the population be expected to be after 25 years?

Ideally a nonlinear regression technique would be used to find the α and β that absolutely minimize the sum of the squares of the errors between the data points and the fitted curve.

A good (but not perfect) answer can be obtained more simply by transforming the data and using linear regression.

The basic procedure:

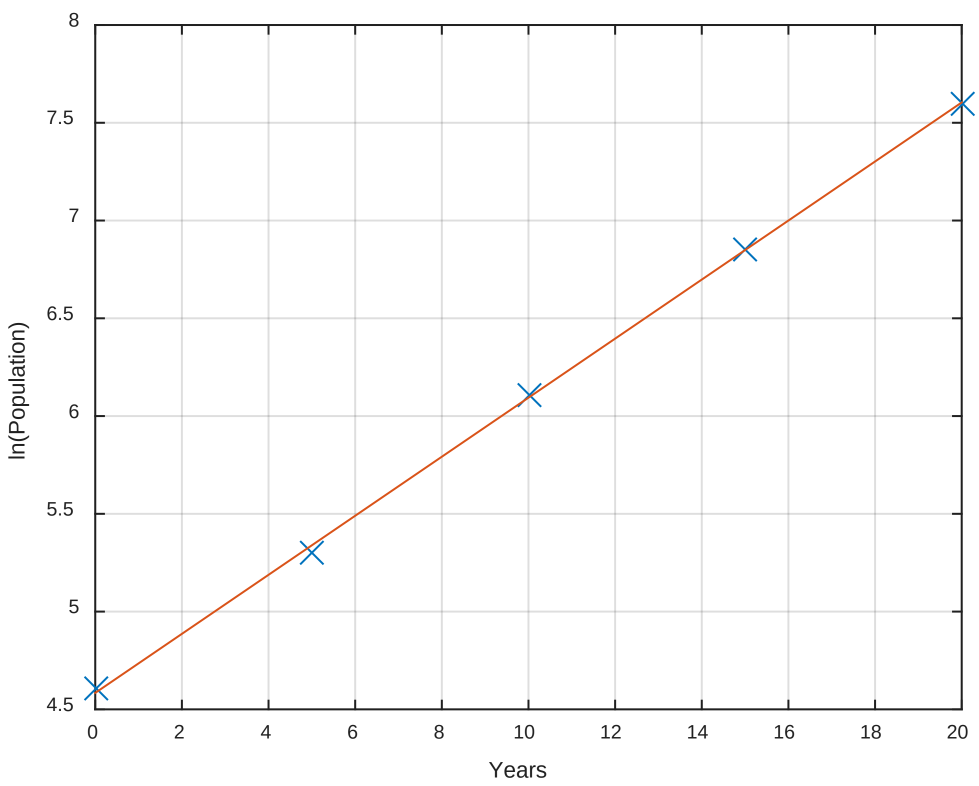
- $y = \alpha * \exp(\beta x) \rightarrow \ln(y) = \ln(\alpha) + \beta x$
- Let $y' = \ln(y)$
- Then $y' = ax + b$, where $a = \beta$ and $b = \ln(\alpha)$
- Use linear regression to find best *a* and *b*.
- Then find α and β by applying $\alpha = \exp(b)$ and $\beta = a$

Matlab part 1:

```
x = [0 5 10 15 20];
y = [100 200 450 950 2000];

yt = log(y);  % transform the y values

p = polyfit (x, yt, 1);  % fit a straight line to the transformed data

fitt = @(x) p(1) * x + p(2);  % function for the fitted line
```

Matlab part 2:

```
figure (1)
plot (x, yt, 'x', x, fitt(x), 'MarkerSize', 10);
grid on; xlabel ('Years'); ylabel ('ln(Population)');

fprintf ('For transformed data a = %f, b = %f, r = %f\n', ...
        p(1), p(2), correlate (x, yt, fitt));
```



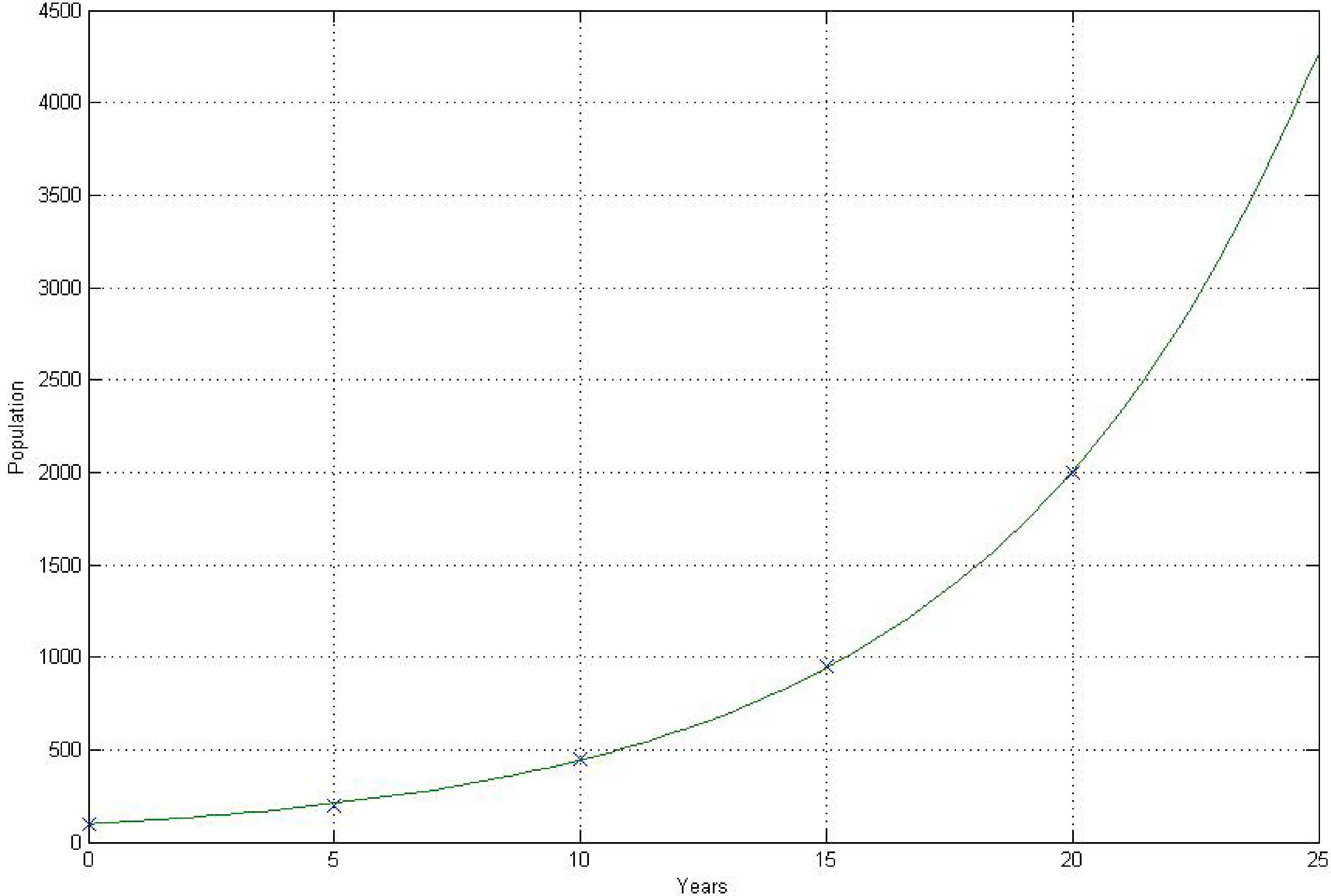
The best fit line is

$y = 0.1510 * x + 4.5841$

The correlation coefficient for this straight line and the transformed data is 0.999789.

Matlab part 3:

```
% calculate alpha and beta
alpha = exp(p(2));
beta = p(1);

fit = @(x) alpha * exp(beta * x); % function for fitted curve

% need lots of x values to get a smooth plot of the fitted curve
xplot = linspace (0, 25, 100);  % plot up to 25 years
yplot = fit(xplot);

figure (2)
plot (x, y, 'x', xplot, yplot, 'MarkerSize', 10);
grid on;
xlabel ('Years');
ylabel ('Population');

fprintf ('For original data alpha = %f, beta = %f, r = %f\n', ...
          alpha, beta, correlate (x, y, fit));

fprintf (‘Predicted population after 25 years = %f\n’, fit(25));
```



This time the first data point does show up.

The best fit curve is y = 97.9148 * exp (0.1510 * x)

The correlation coefficient for this curve and the original data is 0.999957

The predicted population after 25 years is 4268.

The basic idea can be adapted to power equations:

- $y = \alpha x^{\beta} \rightarrow \log(y) = \log(\alpha) + \beta \log(x)$
- Let $x' = \log(x)$ and $y' = \log(y)$
- Then $y' = ax' + b$, where $a = \beta$ and $b = \log(\alpha)$
- Use linear regression to find best a and b.
- Then find α and β by applying $\alpha = 10^{b}$ and $\beta = a$

Note: *log* is used instead of *ln* only for consistency with the text.
ln would work equally well (use $\alpha = \exp(b)$)

And to saturation growth rate equations as well:

- $y = \alpha(x / (\beta + x)) \rightarrow 1/y = (\beta/\alpha)(1/x) + (1/\alpha)$
- Let $x' = 1/x$ and $y' = 1/y$
- Then $y' = ax' + b$, where $a = \beta/\alpha$ and $b = 1/\alpha$
- Use linear regression to find best a and b.
- Then find α and β by applying $\alpha = 1/b$ and $\beta = a/b$

The mathematics of linear regression:

$$\text{Given}: \ (x_1, y_1), (x_2, y_2), (x_3, y_3) \ldots (x_n, y_n)$$

$$\text{To find}: \ \text{the straight line } (y = ax + b) \text{ that best fits the data}$$

$$\text{We must minimize } E = \sum_{i=1}^{n} \left(y_i - (ax_i + b) \right)^2$$

$$= \sum_{i=1}^{n} \left(a^2 x_i^{\ 2} + b^2 + y_i^{\ 2} + 2abx_i - 2ax_i y_i - 2by_i \right)$$

$$\text{At the minimum}:$$

$$\frac{\partial E}{\partial a} = \sum_{i=1}^{n} \left(2ax_i^{\ 2} + 2bx_i - 2x_i y_i \right) = 0$$

$$\frac{\partial E}{\partial b} = \sum_{i=1}^{n} \left(2b + 2ax_i - 2y_i \right) = 0$$

Dividing both equations by 2 and expressing them in matrix form gives:

$$\begin{bmatrix} \sum x_i^{\ 2} & \sum x_i \\ \sum x_i & \sum(1) \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} \sum x_i y_i \\ \sum y_i \end{bmatrix} \quad \text{where } \sum(1) = \sum_{i=1}^{n}(1) = n$$

Solving using Cramer's Rule produces:

$$a = \frac{\begin{vmatrix} b_1 & a_{12} \\ b_2 & a_{22} \end{vmatrix}}{|A|} = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{n\sum x_i^{\ 2} - \left(\sum x_i\right)^2}$$

$$b = \frac{\begin{vmatrix} a_{11} & b_1 \\ a_{21} & b_2 \end{vmatrix}}{|A|} = \frac{\sum x_i^{\ 2} \sum y_i - \sum x_i \sum x_i y_i}{n\sum x_i^{\ 2} - \left(\sum x_i\right)^2}$$

Aside: *b* is more easily calculated using $b = \bar{y} - a\bar{x}$

Calculating of *a* and *b* involves first passing through the data points and calculating the following summations:

$$\sum x_i \quad \sum y_i \quad \sum x_i^{\ 2} \quad \sum x_i y_i$$

Once this is done formulas for *a* and *b* can be applied.

For linear regression ONLY, the correlation coefficient *r* can be computed using:

$$r = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n\sum x_i^{\ 2} - \left(\sum x_i\right)^2}\sqrt{n\sum y_i^{\ 2} - \left(\sum y_i\right)^2}}$$

In addition to the summations listed above this requires $\sum y_i^{\ 2}$

Linear Regression and the Casio Calculator:

formula is y = Ax+B

Mode Mode 2 (REG)	REG stands for regression
1 (LIN)	LIN stands for linear
SHIFT CLR 1 (Scl) =	clear statistical memory
x_1 , y_1 DT	the DT key is the M+ key
x_2 , y_2 DT	
….	and so on until all points entered

To retrieve value of A:

SHIFT S-VAR -> -> 1 (A) = the S-VAR key is the 2 key, -> is right arrow

To retrieve value of B:

SHIFT S-VAR -> -> 2 (B) =

To retrieve the correlation coefficient:

SHIFT S-VAR -> -> 3 (r) =

Other forms of regression are also supported.

Polynomial regression:

Linear regression involves fitting a first order polynomial (i.e. a polynomial of the form $ax + b$) to a set of data points.

The basic idea is readily extended to higher order polynomials.

Example:

```
X:      0      3      6      9     12     15      18      21
Y: 189.4   95.1   34.1    1.8    7.3   46.7   131.9   253.2
```

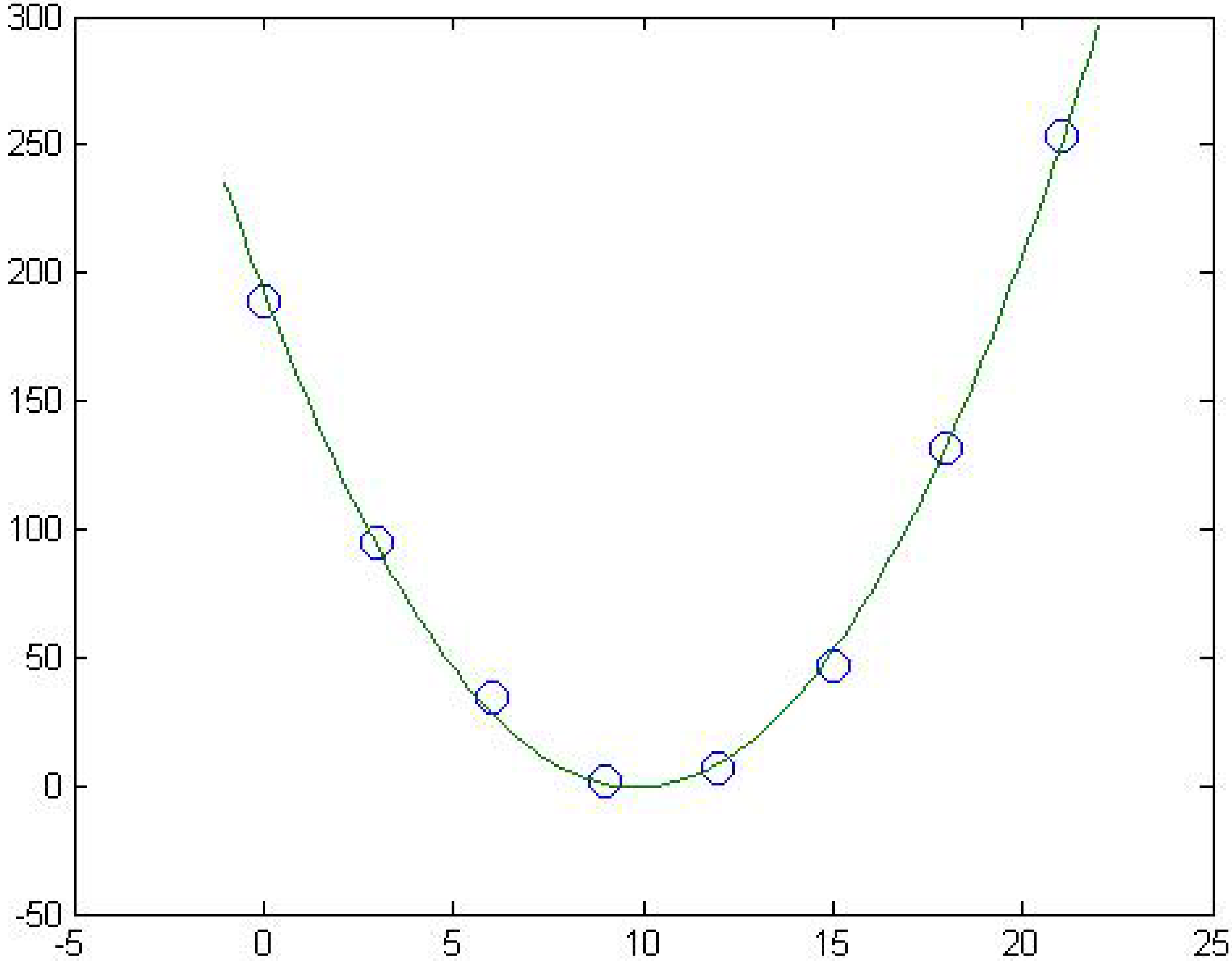
We want to fit a quadratic (i.e. a polynomial of the form $y = ax^2+bx+c$) to the data.

This can be done by using *polyfit* and specifying a second order polynomial.

\>> p = polyfit (x, y, 2)   % 2 for second order

The result is a 3 element containing *a*, *b*, and *c* (in that order).

```
>> xplot = linspace (-1, 22, 100);
>> yplot = polyval (p, xplot);
>> plot (x, y, 'o', xplot, yplot, 'MarkerSize', 10);
```



```
>> fprintf ('The best fit curve is %6.4f * x^2 + %6.4f * x + %6.4f\n',...
              p(1), p(2), p(3));
```

The best fit curve is 2.0088 * x^2 + -39.5105 * x + 193.4125

```
>> f = @(x) p(1) * x .^ 2 + p(2) * x + p(3);
>> r = correlate (x, y, f);
>> fprintf ('The correlation coefficient is %6.4f\n', r);
```

The correlation coefficient is 0.9991

The mathematics of quadratic regression:

Given: $(x_1, y_1), (x_2, y_2), (x_3, y_3) \ldots (x_n, y_n)$

To find: the quadratic ($y = ax^2 + bx + c$) that best fits the data

We must minimize $E = \sum_{i=1}^{n} \left(y_i - (ax_i^{\,2} + bx_i + c) \right)^2$

At the minimum $\frac{\partial E}{\partial a} = \frac{\partial E}{\partial b} = \frac{\partial E}{\partial c} = 0$

First order equations

Filling in the details gives:

$$\begin{bmatrix} \sum x_i^{\,4} & \sum x_i^{\,3} & \sum x_i^{\,2} \\ \sum x_i^{\,3} & \sum x_i^{\,2} & \sum x_i \\ \sum x_i^{\,2} & \sum x_i & \sum (1) \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} \sum x_i^{\,2} y_i \\ \sum x_i y_i \\ \sum y_i \end{bmatrix} \quad \text{where } \sum (1) = \sum_{i=1}^{n} (1) = n$$

The values of *a*, *b*, and *c* can be found by solving this series of equations.
Equations = first order equations plus extra row and column.
This pattern extends to higher order polynomials.