Regression and Curve Fitting

- Regression
- Least-Squares Method
  - Derivation
  - Consistency
  - Bias, Variance
- Polynomial regression and curve fitting

Modelling from data

- The activity of modelling
- Theoretical, experimental, combined modelling
- Experimental modelling: making model from input/output data
- Statistics and machine learning: regression
- Static, dynamic models

Modelling from data

- Model purpose
- Collect empirical data
- Pre-process data
- Optimisation/learning of parameters
- Model verification
- Model validation

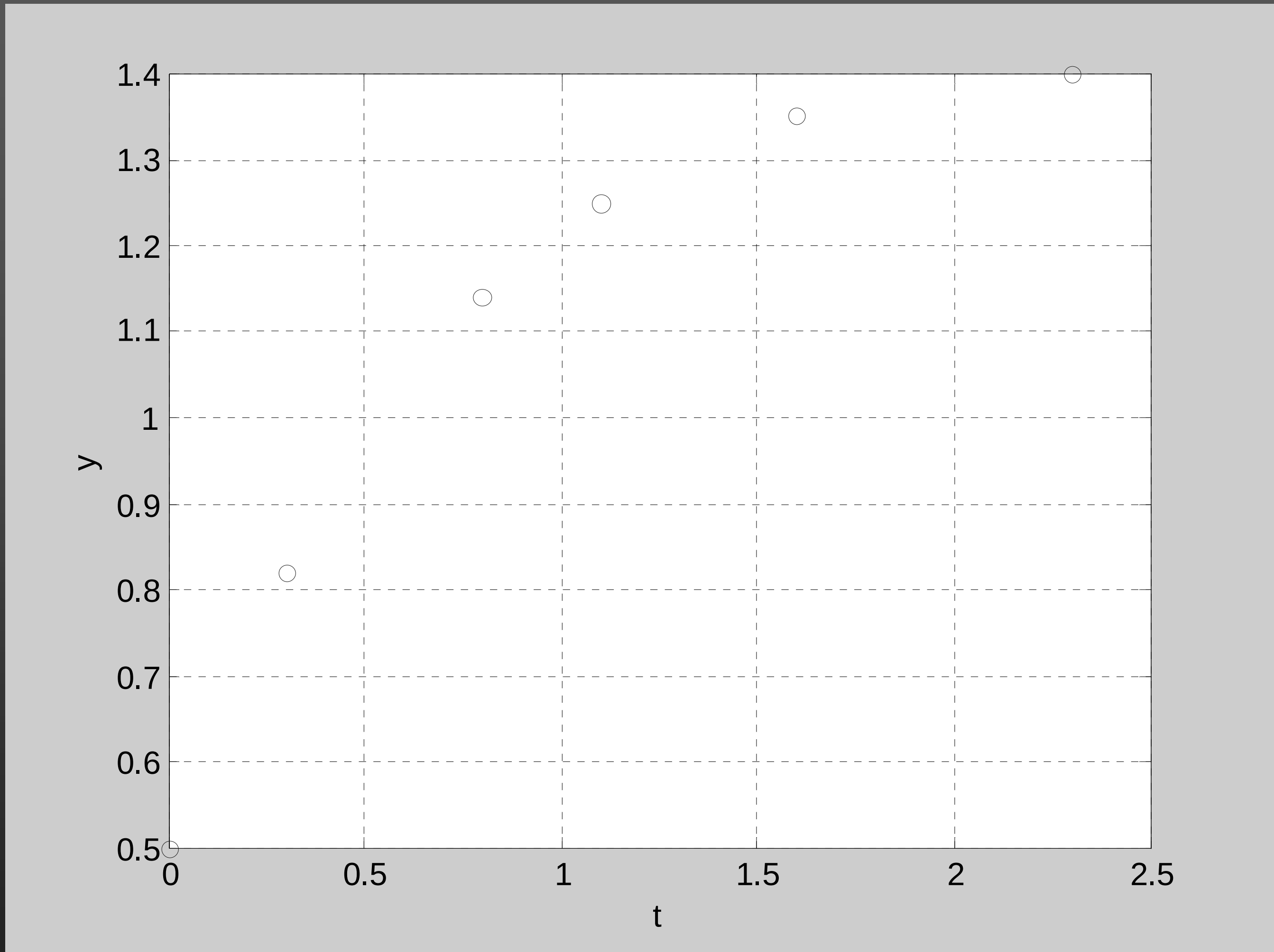
Building a model

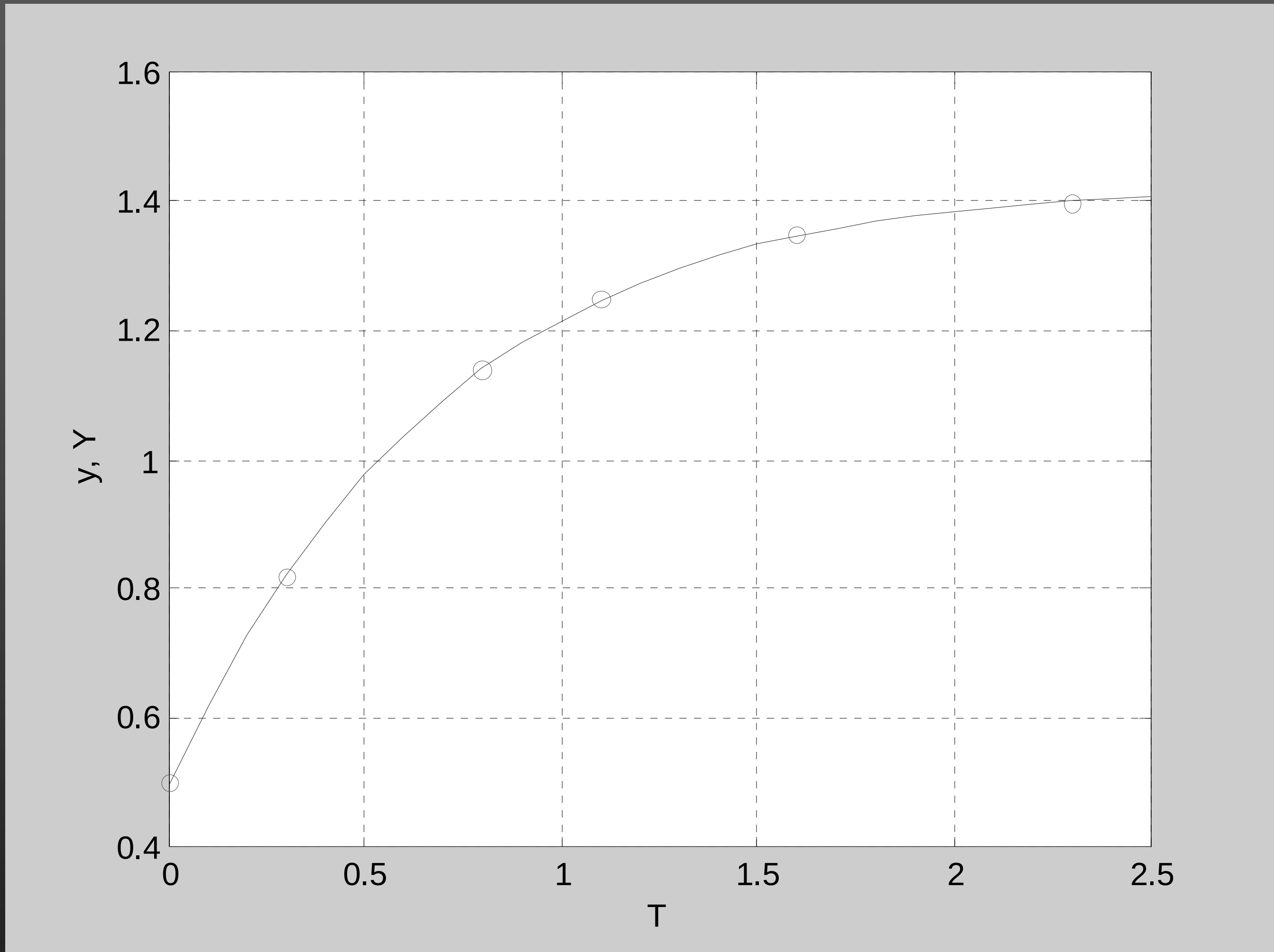
- Regression
  - Least-squares method
  - Bayesian methods
  - Least absolute deviations
  - Nonparametric regression
  - Etc.

Example

t =
0
0.3000
0.8000
1.1000
1.6000
2.3000
y =
0.5000
0.8200
1.1400
1.2500
1.3500
1.4000

$$y = a_0 + a_1 e^{-t} + a_2 t e^{-t}$$


1.4
1.3
1.2
1.1
1
0.9
0.8
0.7
0.6
0.5
y
0
0.5
1
1.5
2
2.5
t



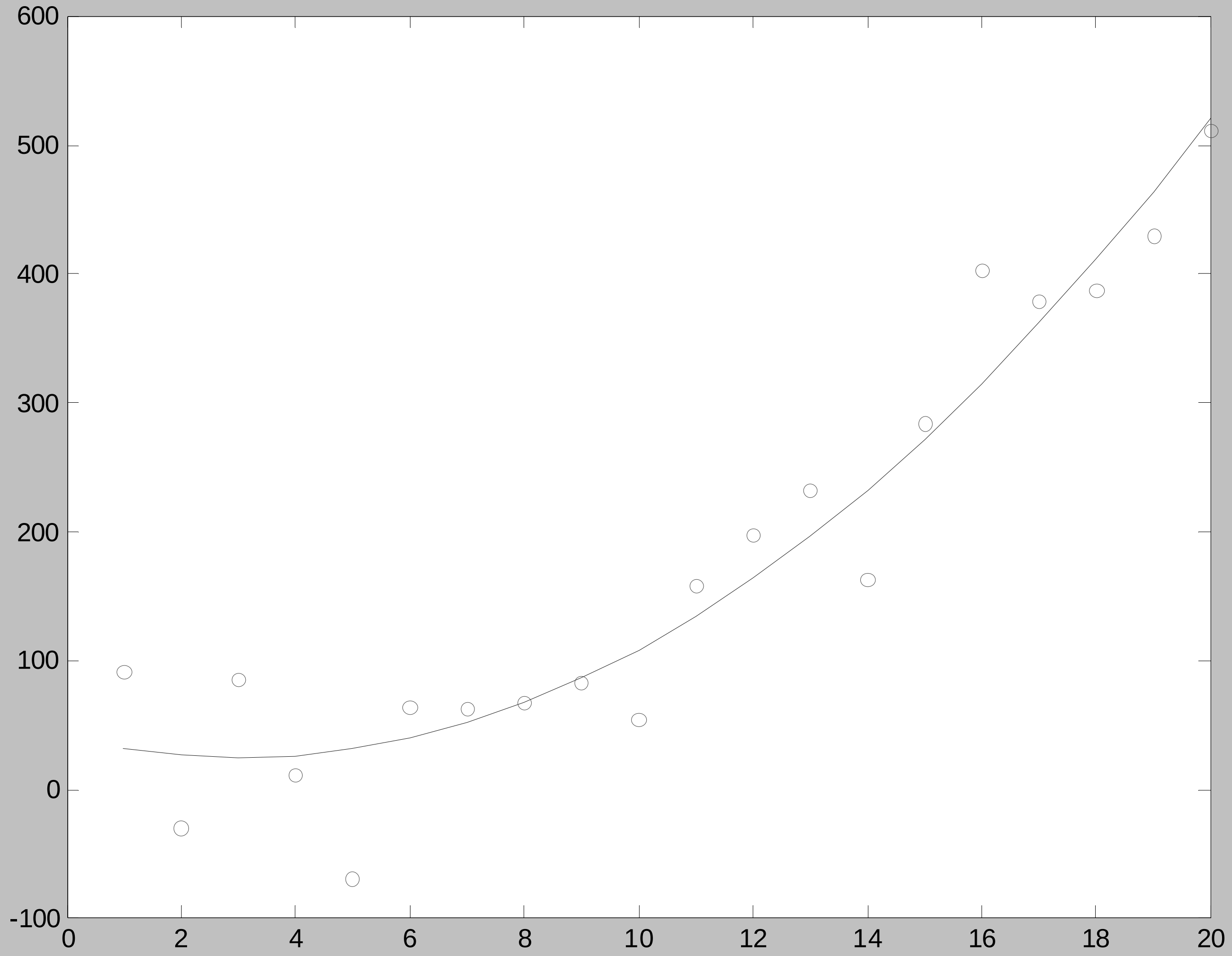
1.6
1.4
1.2
1
0.8
0.6
0.4
0
0.5
1
1.5
2
2.5
y, Y
T

Least-Squares optimisation example

```
% the vector of independent variables
t=(1:20)';
% the vector of measurements
y=[90.446583; - 30.140131;  85.046758;  10.369283; - 70.011198;
   63.811084;   62.268963;  67.068783;  82.392149;   53.275651;
   157.87428;   197.28608;  231.10799;  162.92439;   283.76352;
   401.70519;   378.5821 ;  387.34015;  429.51228;   510.0997 ];
% we wish to model measurements with quadratic function  y=at^2+bt+c
% first we need to put the data in matrix form y=psi*theta or
% y=[t(i)*t(i) t(i) 1][a b c]' which is convenient for calculation.

% matrix psi
psi=[t.*t,t,ones(20,1)];
% the vector of parametrs theta is calculated by LS method as
% theta=[psi'*psi]^-1*psi'*y, which can be in Matlab simplified with left division
theta=psi\y
% calculation of predictions – approximated function
yap=theta(1)*t.*t+theta(2)*t+theta(3);
% figure of measurements (circles) and approximated functions
plot(t, yap,t, y,'o')
```

Least-Squares optimisation example



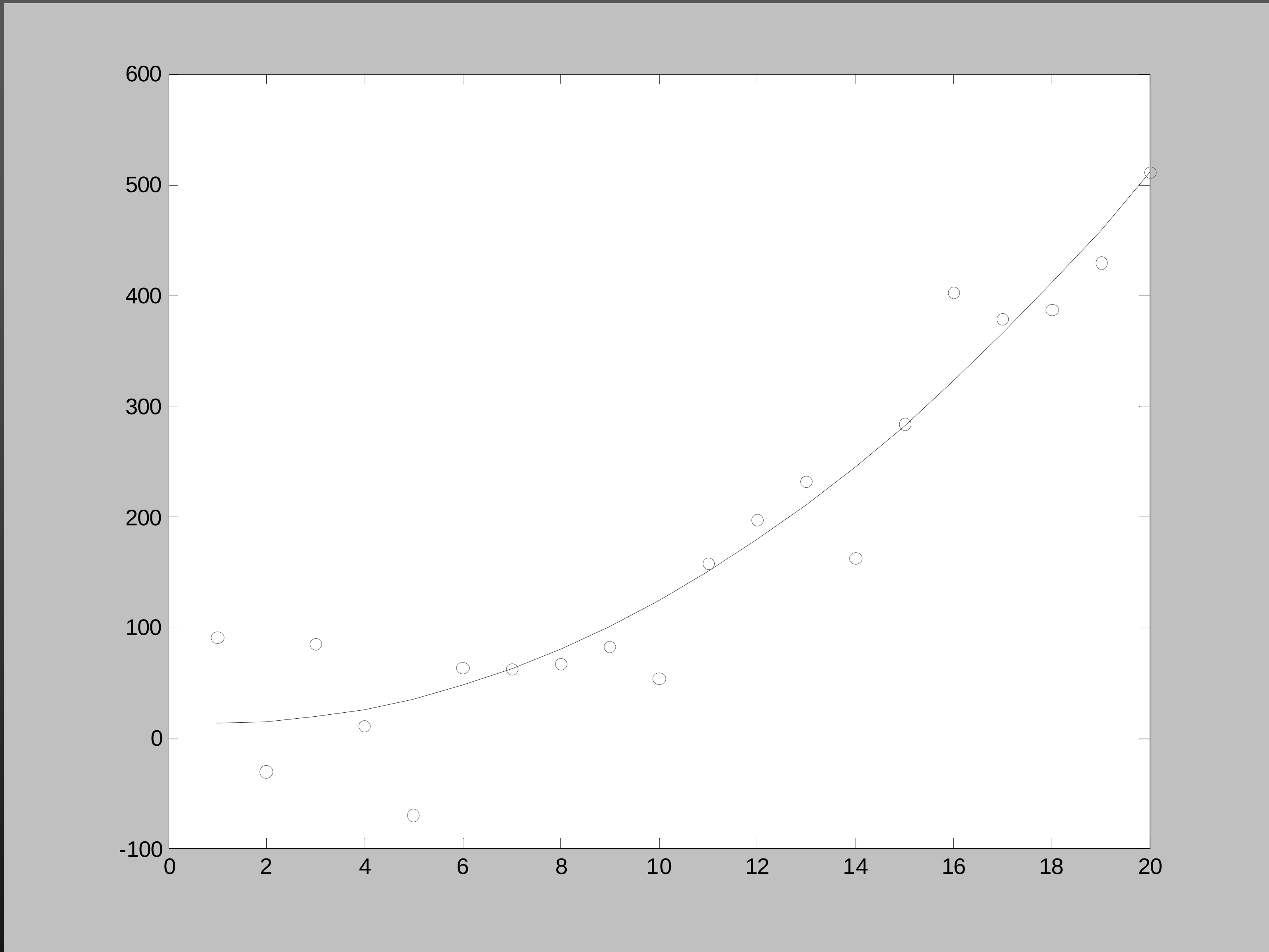
Curve fitting – output error example

```
% the vector of independent variables
t=(1:20)';
% the vector of measurements
y=[90.446583; - 30.140131;  85.046758;  10.369283; - 70.011198;
   63.811084;   62.268963; 67.068783;  82.392149;   53.275651;
   157.87428;   197.28608; 231.10799;  162.92439;   283.76352;
   401.70519;   378.5821 ; 387.34015;  429.51228;   510.0997 ];
% we wish to model measurements with quadratic function  y=at^2+bt+c
% witm minimisation of output error between measurements and
% predicted values.

% the function to which parameters will be fitted on output error
theta=fminsearch(@(theta)sum(abs(y-theta(1)*t.*t-theta(2)*t-theta(3))),[10;10;10])
% calculation of predictions – approximated function
yap=theta(1)*t.*t+theta(2)*t+theta(3);
% figure of measurements (circles) and approximated functions

plot(t, yap,t, y,'o')
```

Curve fitting – output error example



Validation

- Validation on identification\learning data
- Cross-validation on validation\test data

Residuals validation

- Mean absolute error $MAE = \frac{1}{n}\sum_{i=1}^{n}\left|f_i - y_i\right|$
- Mean squared error $MSE = \frac{1}{n}\sum_{i=1}^{n}\left(f_i - y_i\right)^2$
- Mean relative square error $MRSE = \sqrt{\frac{\sum_{i=1}^{n}\left(f_i - y_i\right)^2}{\sum_{i=1}^{n}\left(f_i\right)^2}}$
- Log predictive density

$$LPD = \frac{1}{2n}\sum_{i=1}^{n}\left(\log(2\pi) + \log(\sigma^2) + \frac{(f_i - y_i)^2}{\sigma^2}\right)$$