Rutgers University
School of Engineering

Fall 2011

14:440:127 - Introduction to Computers for Engineers

Sophocles J. Orfanidis
ECE Department
orfanidi@ece.rutgers.edu

week 12

Weekly Topics

Week 1 - Basics – variables, arrays, matrices, plotting (ch. 2 & 3)
Week 2 - Basics – operators, functions, program flow (ch. 2 & 3)
Week 3 - Matrices (ch. 4)
Week 4 - Plotting – 2D and 3D plots (ch. 5)
Week 5 - User-defined functions (ch. 6)
Week 6 - Input-output processing (ch. 7)
Week 7 - Program flow control & relational operators (ch. 8)
Week 8 - Matrix algebra – solving linear equations (ch. 9)
Week 9 - Strings, structures, cell arrays (ch. 10)
Week 10 - Symbolic math (ch. 11)
Week 11 - Numerical methods – data fitting (ch. 12)
Week 12 - Numerical methods – data fitting – part II (ch. 12)

Textbook: H. Moore, *MATLAB for Engineers*, 2nd ed., Prentice Hall, 2009

Numerical Methods
Data Fitting – part II

- data fitting with polynomials – **`polyfit, polyval`**
- examples: Moore's law,
- Hank Aaaron,
- US census data
- least-squares polynomial regression
- least-squares with other basis functions
- examples: exponential models
- trigonometric basis functions
- trigonometric with polynomial trends (CO2 data)

Polynomial data fitting - review

polyfit, polyval

$$P(x) = p_1 x^M + p_2 x^{M-1} + \cdots + p_M x + p_{M+1}$$

$$\mathbf{p} = [p_1,\ p_2, \ldots,\ p_M,\ p_{M+1}]$$

```
>> doc polyfit
>> doc polyval
```

$$P(x) = 5x^4 - 2x^3 + x^2 + 4x + 3$$

$$\mathbf{p} = [5,\ -2,\ 1,\ 4,\ 3]$$

polynomial $P(x)$ is represented by its coefficients $\mathbf{p}$

Given N data points $\{x_i, y_i\}$, $i=1,2,\ldots,N$, find an M-th degree polynomial that best fits the data – **(polyfit)**

```
% design procedure:

xi = [x1,x2,...,xN];
yi = [y1,y2,...,yN];

p = polyfit(xi,yi,M);

y = polyval(p,x);
```

evaluate $P(x)$ at a given vector x

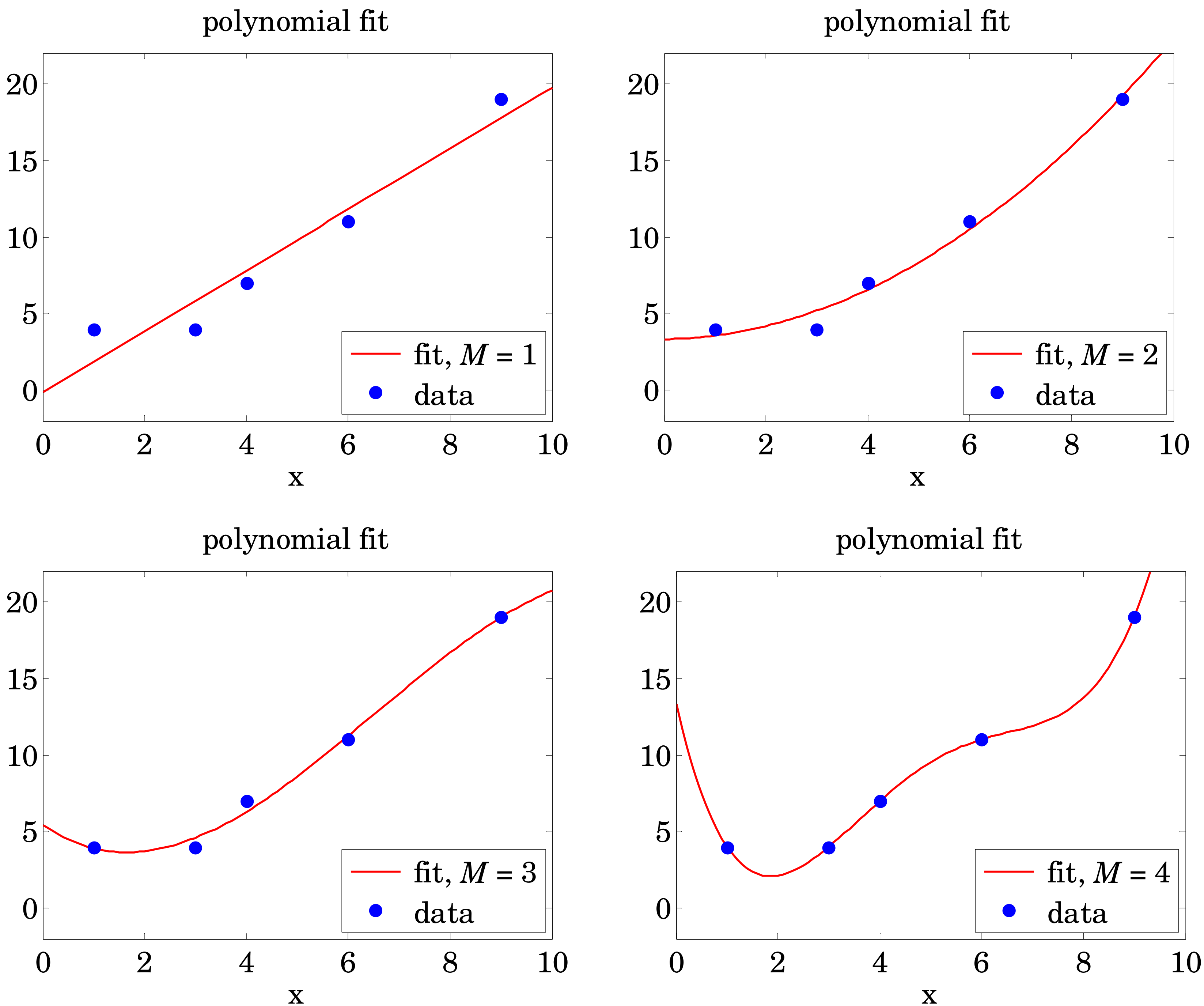
M = polynomial order

if $N = M+1$, the polynomial interpolates the data

if $N > M+1$, the polynomial provides the best fit in a least-squares sense

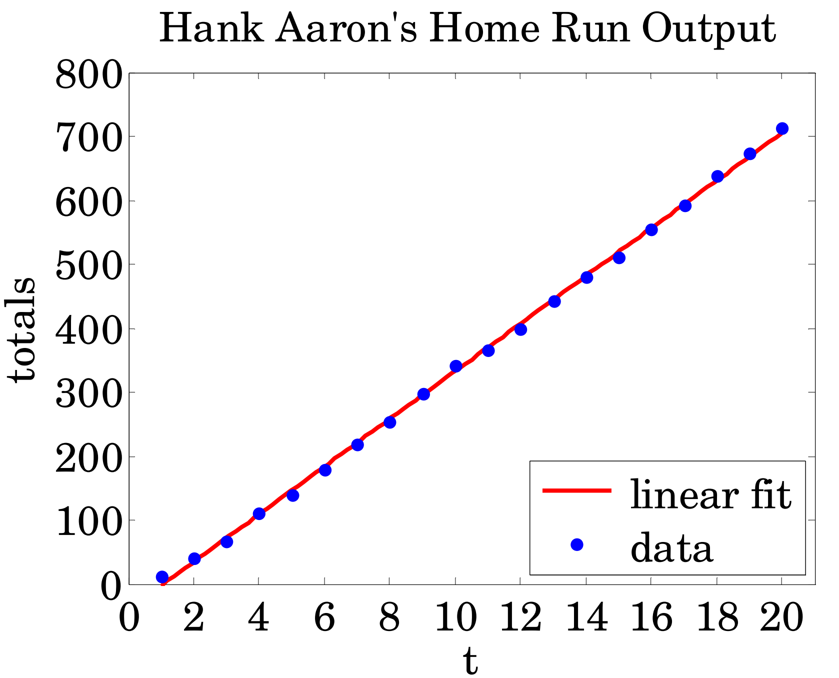
$$J = \sum_{i=1}^{N} \left(P(x_i) - y_i\right)^2 = \min$$

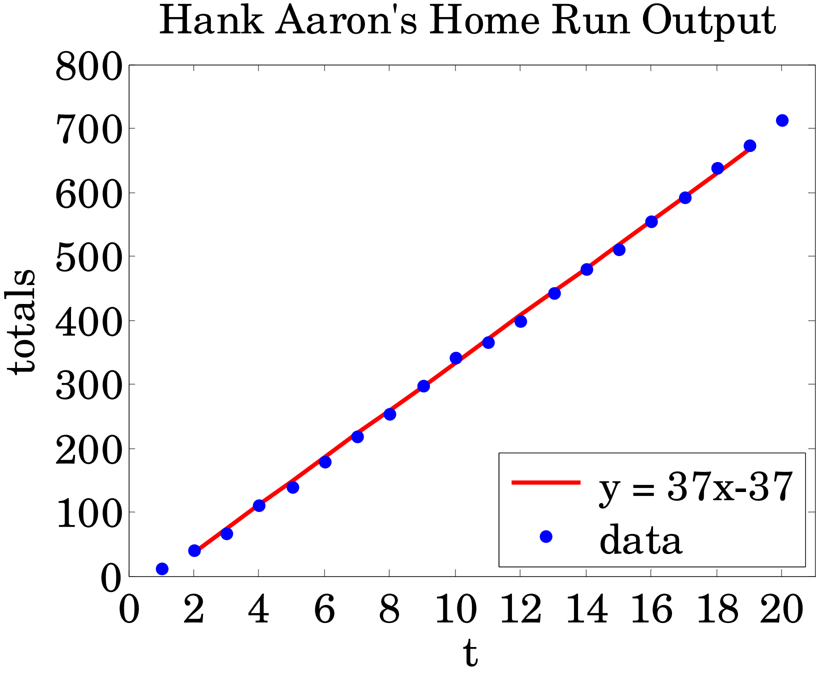
```
xi = [1, 3, 4, 6, 9];
yi = [4, 4, 7, 11, 19];

x = linspace(0,10,101);

for M = [1,2,3,4]
  p = polyfit(xi,yi,M);

  y = polyval(p,x);

  figure;
  plot(x,y,'r-', xi,yi,'b.', 'markersize',25);
  yaxis(-2,22,0:5:20); xaxis(0,10,0:2:10);
  xlabel('x'); title('polynomial fit');
  legend([' fit, {\itM} = ',num2str(M)],...
         ' data', 'location','se');
end
```


polynomial fit
20
15
10
5
0
0
2
4
6
8
10
x
fit, $M = 1$
data
polynomial fit
fit, $M = 2$
data
polynomial fit
fit, $M = 3$
data
polynomial fit
fit, $M = 4$
data

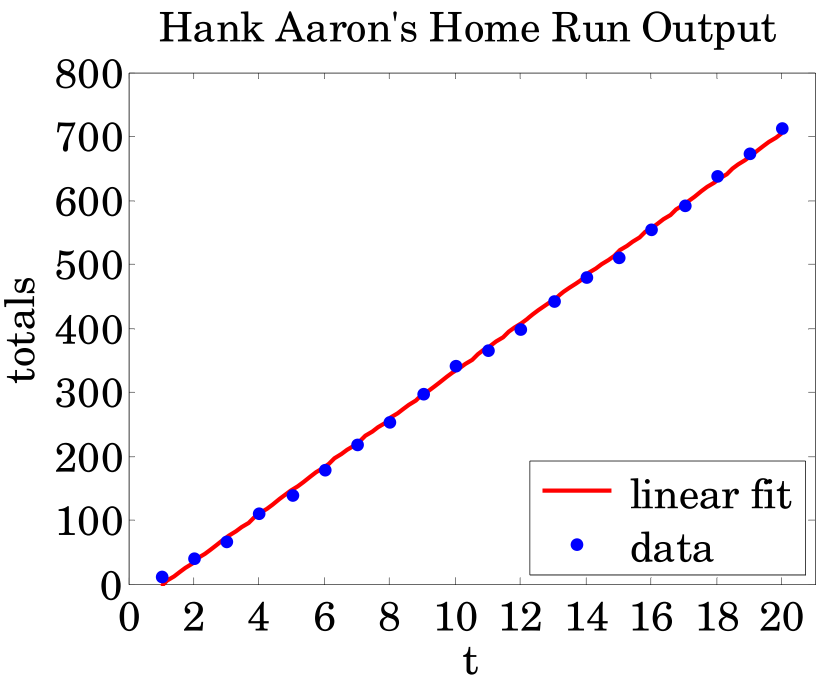
```
% year     ti    H
% ----------------
  1954      1    13
  1955      2    27
  1956      3    26
  1957      4    44
  1958      5    30
  1959      6    39
  1960      7    40
  1961      8    34
  1962      9    45
  1963     10    44
  1964     11    24
  1965     12    32
  1966     13    44
  1967     14    39
  1968     15    29
  1969     16    44
  1970     17    38
  1971     18    47
  1972     19    34
  1973     20    40
```

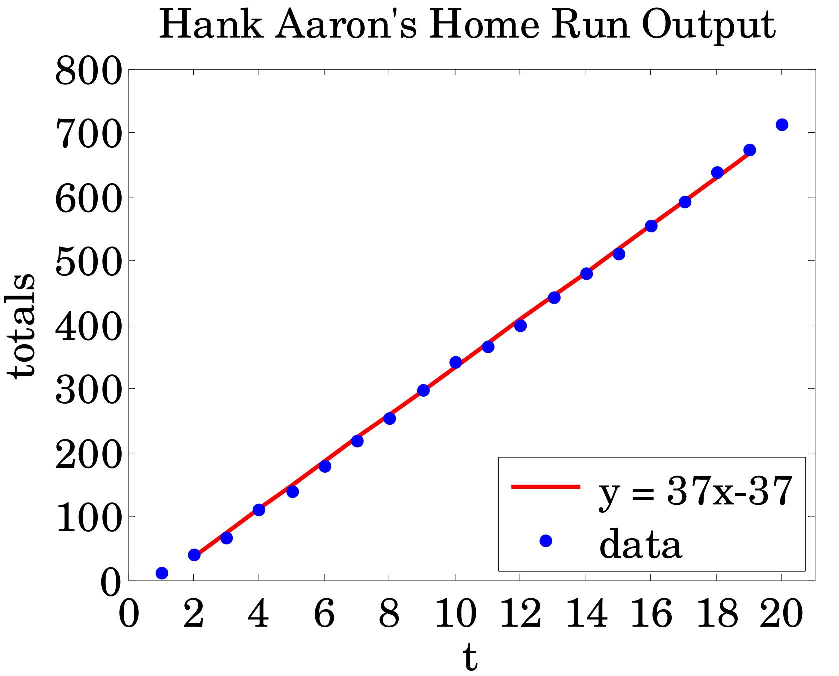
from set-6





```
A = load('aaron.dat');

ti = A(:,2); H = A(:,3);
yi = cumsum(H);

p = polyfit(ti,yi,1)

p =
   37.2617  -39.8474

t = linspace(1,20, 101);
y = polyval(p,t);

plot(t,y,'r-', ...
     ti,yi,'b.', ...
     'markersize', 18);
```





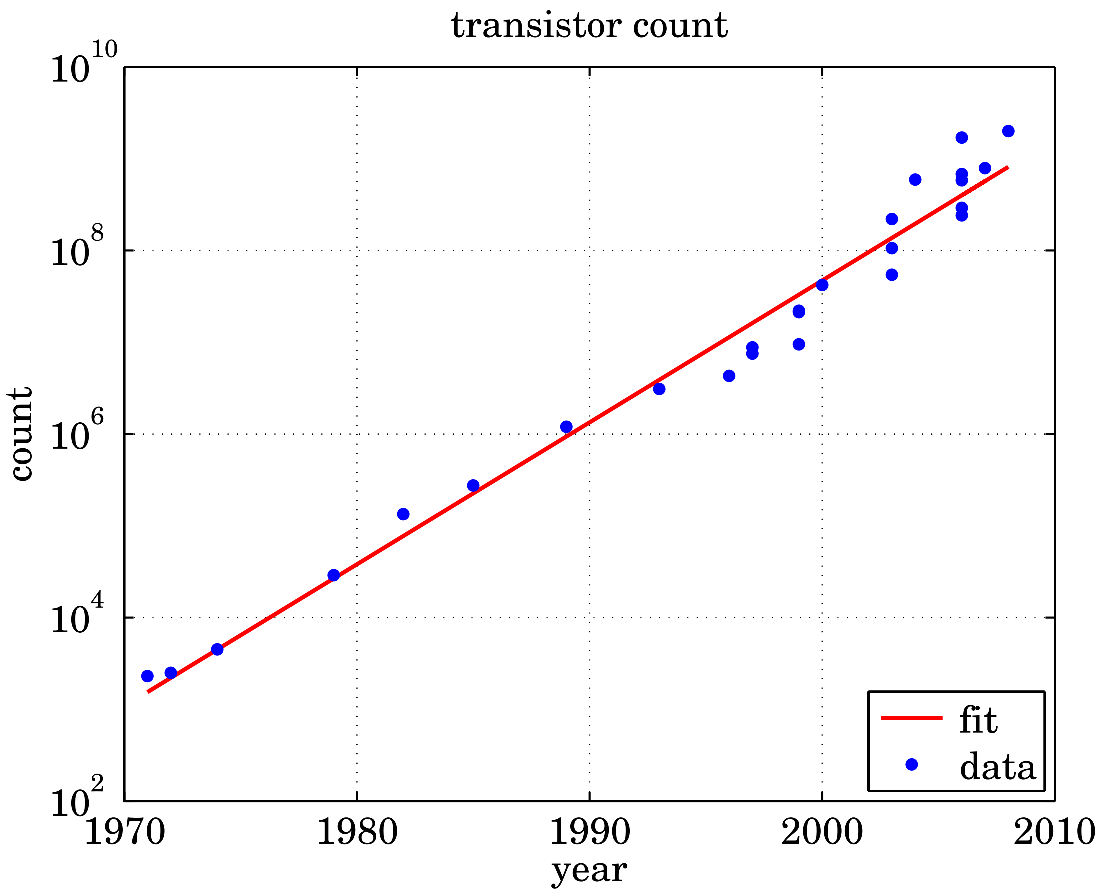
Given N data points $\{x_i, y_i\}$, $i=1,2,\ldots,N$, the following data models can be reduced to linear fits using an appropriate transformation of the data:

linear:	$y = a\,x + b$		
exponential:	$y = b\,e^{ax}$	$\Rightarrow$	$\log(y) = ax + \log(b)$
exponential:	$y = b\,2^{ax}$	$\Rightarrow$	$\log_2(y) = ax + \log_2(b)$
exponential:	$y = b\,x\,e^{ax}$	$\Rightarrow$	$\log(y/x) = ax + \log(b)$
power:	$y = b\,x^{a}$	$\Rightarrow$	$\log(y) = a\,\log(x) + \log(b)$

```
>> p = polyfit(xi,log(yi),1);   % exponential
>> y = exp(polyval(p,x));   % y=exp(a*x+log(b))
>> a = p(1);
>> b = exp(p(2));     % so that y = b*exp(a*x)
```

yi	ti
2.300e+003	1971
2.500e+003	1972
4.500e+003	1974
2.900e+004	1979
1.340e+005	1982
2.750e+005	1985
1.200e+006	1989
3.100e+006	1993
4.300e+006	1996
7.500e+006	1997
8.800e+006	1997
9.500e+006	1999
2.130e+007	1999
2.200e+007	1999
4.200e+007	2000
5.430e+007	2003
1.059e+008	2003
2.200e+008	2003
5.920e+008	2004
2.410e+008	2006
2.910e+008	2006
5.820e+008	2006
6.810e+008	2006
7.890e+008	2007
1.700e+009	2006
2.000e+009	2008

Moore's law from set-4



fitted model:

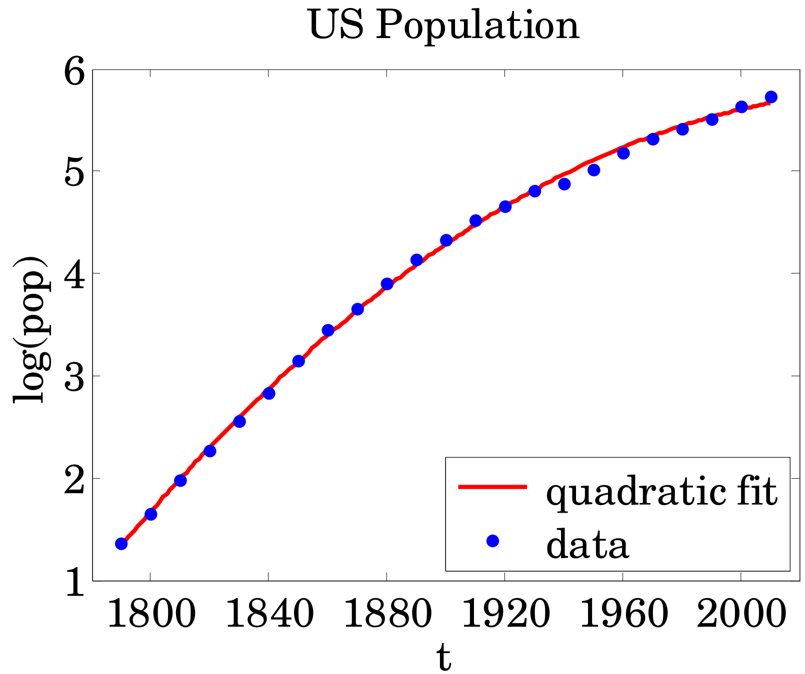
```
f(t) = b*2.^(a*(t-t1));
```

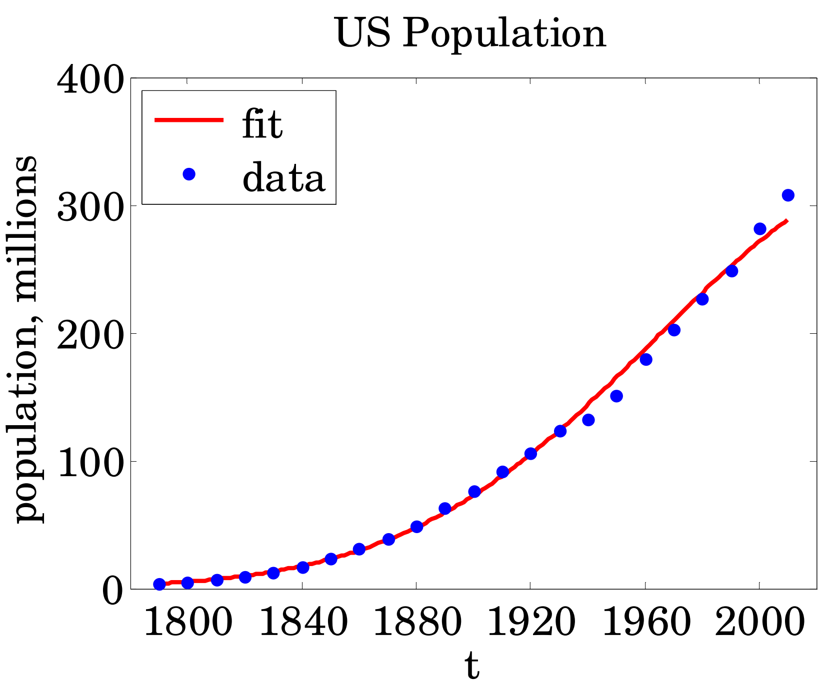
```
Y = load('transistor_count.dat');

y = Y(:,1);  t = Y(:,2);

t1 = t(1);

p = polyfit(t-t1, log2(y), 1);

p =
    0.5138  10.5889    % b = 2^p(2) = 1.5402e+003

f = 2.^(polyval(p,t-t1));

semilogy(t,f,'r-', t,y,'b.', 'markersize',18)
```

fitted model:

```
f(t) = b * 2.^(a*(t-t1)) = 2.^(a*(t-t1)+log2(b));

% a = p(1), log2(b) = p(2) --> b = 2^(p(2))
```

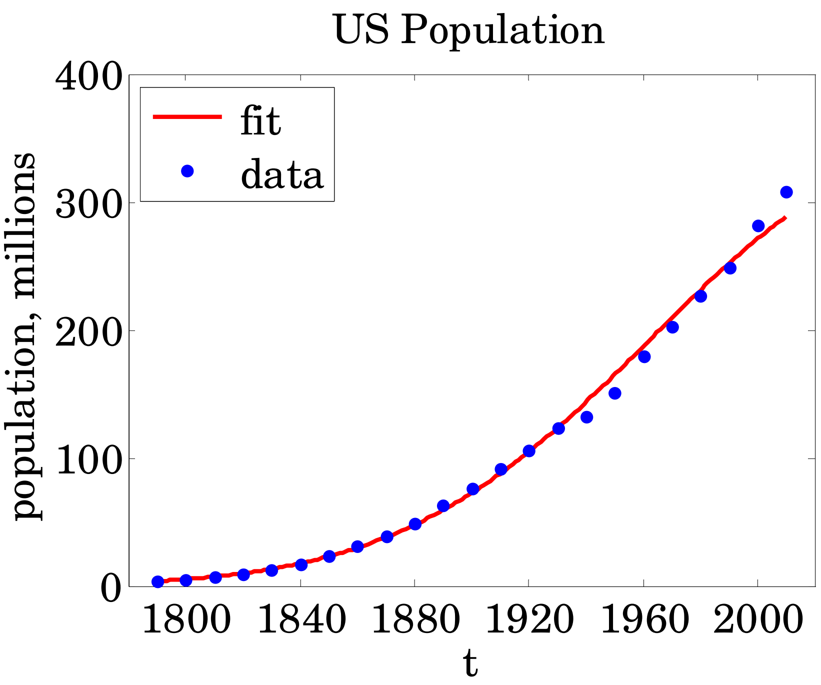
```
% source: Wikipedia
% US population in millions
%
%  ti        yi
% -------------
  1790      3.929
  1800      5.237
  1810      7.240
  1820      9.638
  1830     12.866
  1840     17.069
  1850     23.192
  1860     31.443
  1870     38.558
  1880     49.371
  1890     62.980
  1900     76.212
  1910     92.229
  1920    106.022
  1930    123.202
  1940    132.165
  1950    151.326
  1960    179.323
  1970    203.212
  1980    226.546
  1990    248.710
  2000    281.422
  2010    308.746
```





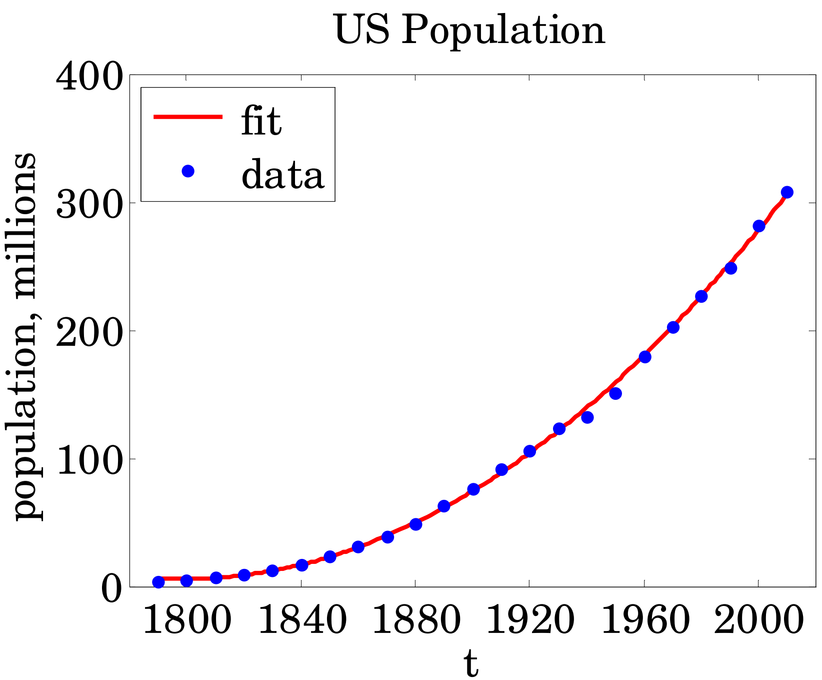
```
A = load('uspop.dat');

ti = A(:,1); yi = A(:,2);

p = polyfit(ti,log(yi),2)     % quadratic fit

p =
   -0.0001    0.2653 -266.4672

t = linspace(1790, 2010, 201);
y = exp(polyval(p,t));

figure; plot(t, log(y), 'r-', ...
             ti,log(yi),'b.','markersize',18);

figure; plot(t, y,'r-', ...
             ti,yi,'b.','markersize',18);
```

```
A = load('uspop.dat');

ti = A(:,1); yi = A(:,2); t1 = ti(1);

p = polyfit(ti,yi,2)                  % quadratic fit
p1 = polyfit(ti-t1,yi,2)              % t1 = 1790

t = linspace(1790, 2010, 201);
y = polyval(p,t);
y1 = polyval(p1,t-t1);               % shifted origin

norm(y-y1)                           % = 7.7100e-011

plot(t, y,'r-', ti,yi,'b.','markersize',18);

>> num2str([p',p1'],'%12.2e')

ans =
 6.78e-003   6.78e-003
-2.44e+001  -1.32e-001
 2.20e+004   6.51e+000
```



exponential fit

polynomial fit



How does **polyfit** work? Consider a straight-line fit, $y = ax+b$, to N data points $\{x_i, y_i\}$, $i=1,2,\ldots,N$

polynomial regression

overdetermined & inconsistent linear system of 5 equations in 2 unknowns

least-squares solution

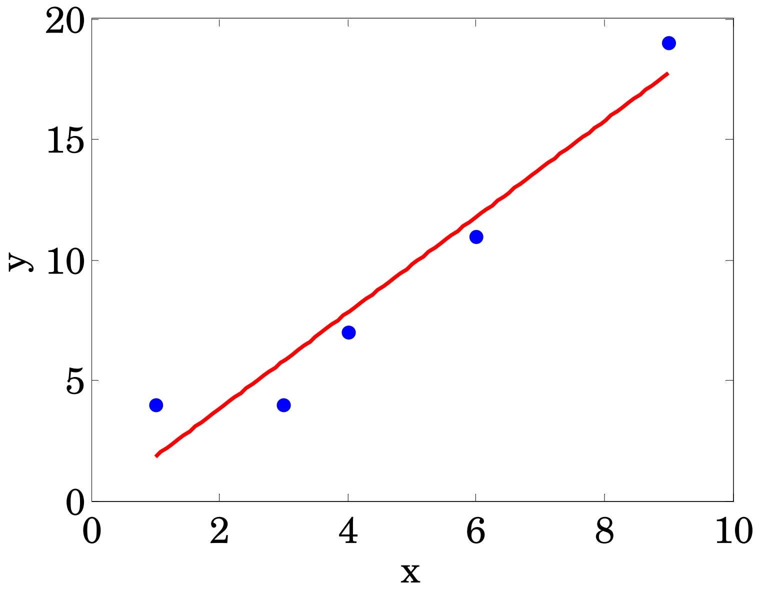
$$
\begin{aligned}
a\,x_1 + b &= y_1 \\
a\,x_2 + b &= y_2 \\
a\,x_3 + b &= y_3 \\
a\,x_4 + b &= y_4 \\
a\,x_5 + b &= y_5
\end{aligned}
\quad \Rightarrow \quad
\begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ x_3 & 1 \\ x_4 & 1 \\ x_5 & 1 \end{bmatrix}
\begin{bmatrix} a \\ b \end{bmatrix}
=
\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix},
\quad \mathbf{p} = \begin{bmatrix} a \\ b \end{bmatrix}
$$

A is the design matrix

$\mathbf{A}\,\mathbf{p} = \mathbf{y}$

$\mathbf{p} = \mathbf{A} \setminus \mathbf{y}$

```
xi = [1, 3, 4, 6, 9]';      % column vectors
yi = [4, 4, 7, 11, 19]';

A = [xi, ones(5,1)];        % design matrix
p = A\yi
p =
    1.9892
   -0.1505

p = polyfit(xi,yi,1)
p =
    1.9892   -0.1505

x = linspace(1,9,91);
y = polyval(p,x);

plot(x,y,'r', xi,yi,'b.','markersize',20);
```



Quadratic fit, $y = ax^2 + bx + c$
to N data points $\{x_i, y_i\}$, $i=1, 2, \ldots, N$

polynomial regression

overdetermined & inconsistent linear system of 5 equations in 3 unknowns

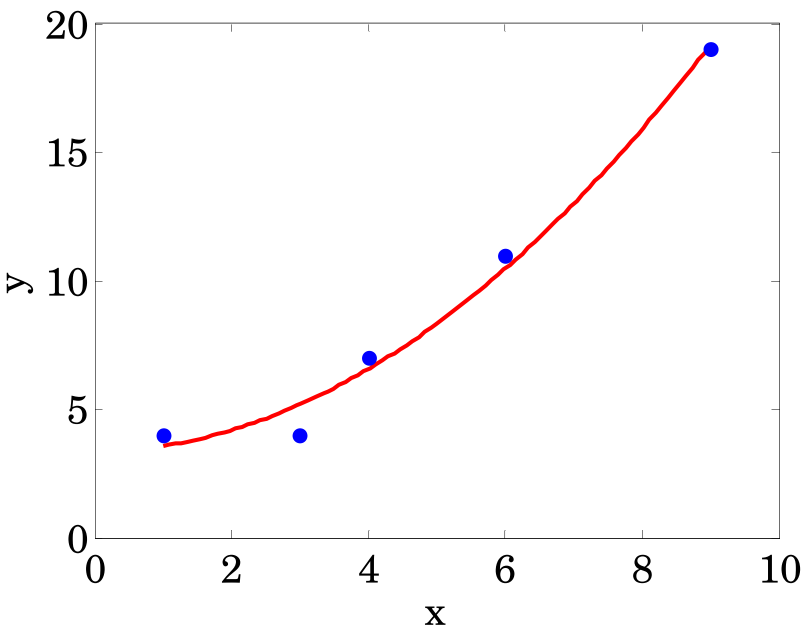
least-squares solution

$$
\begin{aligned}
a\,x_1^2 + b\,x_1 + c &= y_1 \\
a\,x_2^2 + b\,x_2 + c &= y_2 \\
a\,x_3^2 + b\,x_3 + c &= y_3 \\
a\,x_4^2 + b\,x_4 + c &= y_4 \\
a\,x_5^2 + b\,x_5 + c &= y_5
\end{aligned}
\quad \Rightarrow \quad
\begin{bmatrix} x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \\ x_3^2 & x_3 & 1 \\ x_4^2 & x_4 & 1 \\ x_5^2 & x_5 & 1 \end{bmatrix}
\begin{bmatrix} a \\ b \\ c \end{bmatrix}
=
\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix},
\quad \mathbf{p} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}
$$

$$\boxed{\mathbf{p} = \mathbf{A} \setminus \mathbf{y}}$$

$$\mathbf{A}\,\mathbf{p} = \mathbf{y}$$

```
xi = [1, 3, 4, 6, 9]';          % column vectors
yi = [4, 4, 7, 11, 19]';

A = [xi.^2, xi, xi.^0];         % design matrix
p = A\yi
p =
    0.1905
    0.0476
    3.3333

p = polyfit(xi,yi,2)
p =
   0.1905   0.0476   3.3333

x = linspace(1,9,91);
y = p(1)*x.^2 + p(2)*x + p(3);    % polyval(p,x)

plot(x,y,'r', xi,yi,'b.','markersize',20);
```



least-squares solution

Euclidean L_2 norm

polynomial regression

$||\mathbf{A}\,\mathbf{p} - \mathbf{y}||^2 = \min \quad \Rightarrow$ `p = A \ y`

equivalent solutions:

```
p = (A'A)\(A'*y)
p = pinv(A)*y
```

assumes that $M+1 \le N$ and that **A** has full rank, conditions that are typically satisfied in practice (then, **p** is unique least-squares solution)

other norms – such as L_1 – are used in practice but don't have a closed-form solution – several MATLAB toolboxes exist for such problems

The data model is assumed to be a linear combination of known basis functions, such as exponential, trigonometric, etc:

regression with other basis functions

$$y = c_0 + c_1 f_1(x) + c_2 f_2(x) + \cdots + c_M f_M(x)$$

and the objective is to determine the coefficients c_i to fit N data points $\{x_i, y_i\}$, $i = 1, 2, \ldots, N$, where again we must assume $M+1 \le N$

Polynomial fitting is a special case using the monomial basis: 1, x, x^2, …, x^M

Design procedure: set up the design matrix **A** and solve the overdetermined linear system **A c = y**

A c = y
c = A \ y

Example: $M = 3, \quad N = 5$

$$y = c_0 + c_1 f_1(x) + c_2 f_2(x) + c_3 f_3(x)$$

regression with other basis functions

$$\begin{aligned}
c_0 + c_1 f_1(x_1) + c_2 f_2(x_1) + c_3 f_3(x_1) &= y_1 \\
c_0 + c_1 f_1(x_2) + c_2 f_2(x_2) + c_3 f_3(x_2) &= y_2 \\
c_0 + c_1 f_1(x_3) + c_2 f_2(x_3) + c_3 f_3(x_3) &= y_3 \\
c_0 + c_1 f_1(x_4) + c_2 f_2(x_4) + c_3 f_3(x_4) &= y_4 \\
c_0 + c_1 f_1(x_5) + c_2 f_2(x_5) + c_3 f_3(x_5) &= y_5
\end{aligned}$$

$$\underbrace{\begin{bmatrix} 1 & f_1(x_1) & f_2(x_1) & f_3(x_1) \\ 1 & f_1(x_2) & f_2(x_2) & f_3(x_2) \\ 1 & f_1(x_3) & f_2(x_3) & f_3(x_3) \\ 1 & f_1(x_4) & f_2(x_4) & f_3(x_4) \\ 1 & f_1(x_5) & f_2(x_5) & f_3(x_5) \end{bmatrix}}_{\text{design matrix, } \mathbf{A}} \underbrace{\begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}}_{\text{coeffs, } \mathbf{c}} = \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix}}_{\text{data, } \mathbf{y}}$$

A c = y

c = A \ y

Examples of other models reducible to the standard form

$$y = \exp\left(c_0 + c_1 f_1(x) + c_2 f_2(x) + c_3 f_3(x)\right)$$

$$y = \sqrt{c_0 + c_1 f_1(x) + c_2 f_2(x) + c_3 f_3(x)}$$

$$y = \frac{1}{c_0 + c_1 f_1(x) + c_2 f_2(x) + c_3 f_3(x)}$$

$$y = \sqrt{1 + \frac{f(x)}{c_0 + c_1 f_1(x) + c_2 f_2(x) + c_3 f_3(x)}}$$

$$\log(y) = c_0 + c_1 f_1(x) + c_2 f_2(x) + c_3 f_3(x)$$

$$y^2 = c_0 + c_1 f_1(x) + c_2 f_2(x) + c_3 f_3(x)$$

$$\frac{1}{y} = c_0 + c_1 f_1(x) + c_2 f_2(x) + c_3 f_3(x)$$

$$\frac{f(x)}{y^2 - 1} = c_0 + c_1 f_1(x) + c_2 f_2(x) + c_3 f_3(x)$$

ti	yi
0	42.7
1	46.7
2	59.1
3	69.5
4	81.0
5	80.7
6	83.2
7	72.0
8	67.1
9	52.6
10	43.7
11	40.9
12	38.6
13	48.8
14	57.2
15	71.2
16	77.5
17	79.8
18	82.3
19	76.3
20	61.5
21	53.0
22	41.5
23	37.3

regression with other basis functions

Example 1: modeling of temperature variations in a city over 24 months

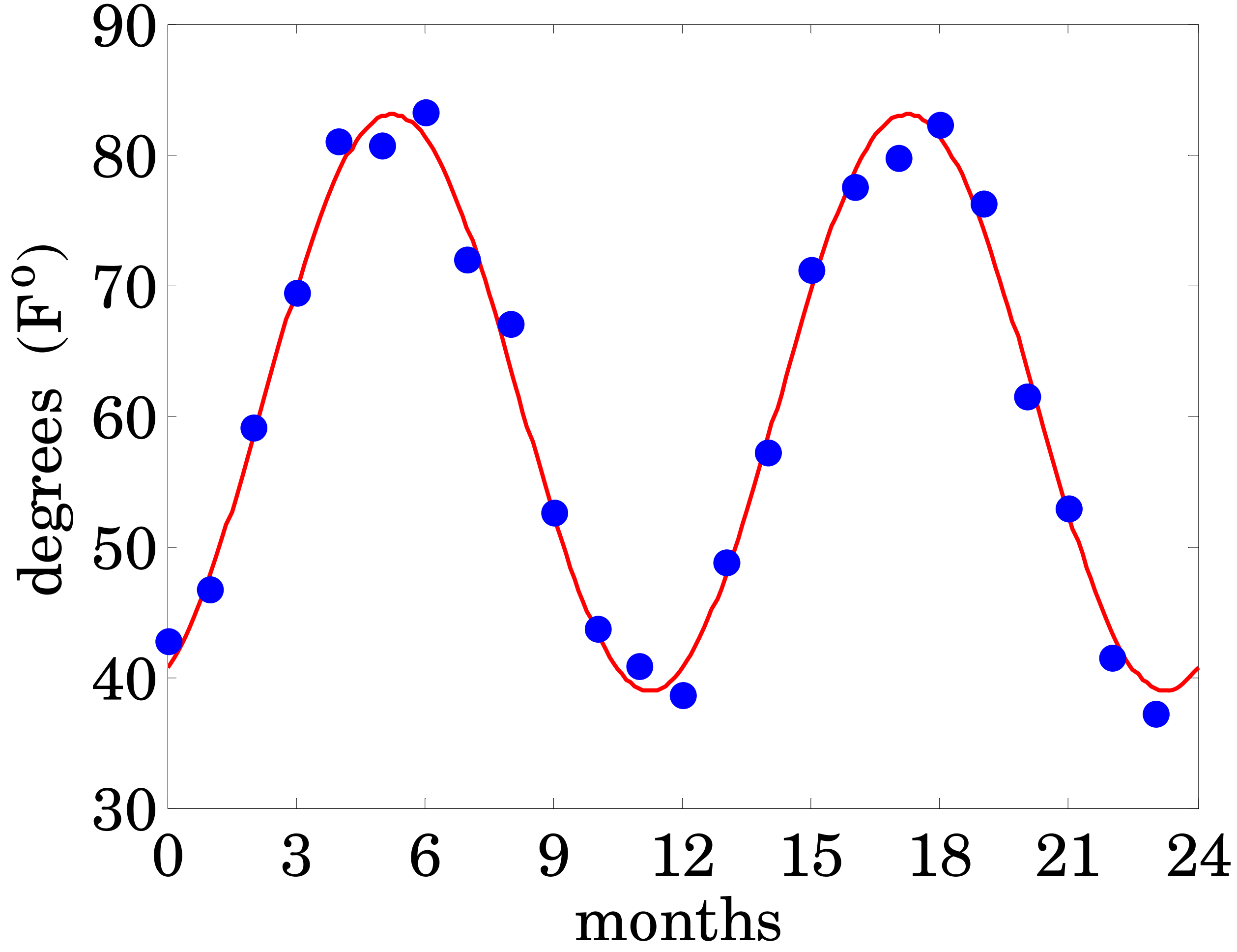
$$y(t) = c_0 + c_1 \underbrace{\cos\left(\frac{2\pi t}{12}\right)} + c_2 \underbrace{\sin\left(\frac{2\pi t}{12}\right)}$$

basis functions

```
N = length(ti);

A = [ones(N,1), cos(2*pi*ti/12), sin(2*pi*ti/12)];

c = A\yi

c =
   61.0083
  -20.3333
    8.5565

f = @(t) c(1) + c(2) * cos(2*pi*t/12) + ...
         c(3) * sin(2*pi*t/12);

t = linspace(0,24,241);

plot(t,f(t),'r', ti,yi,'b.','markersize',25);
```

24x3 design matrix

estimated parameters
estimated model



degrees (F^0)
90
80
70
60
50
40
30
0
3
6
9
12
15
18
21
24
months

Example 2: $y = \frac{c_1}{x} + c_2\, x$

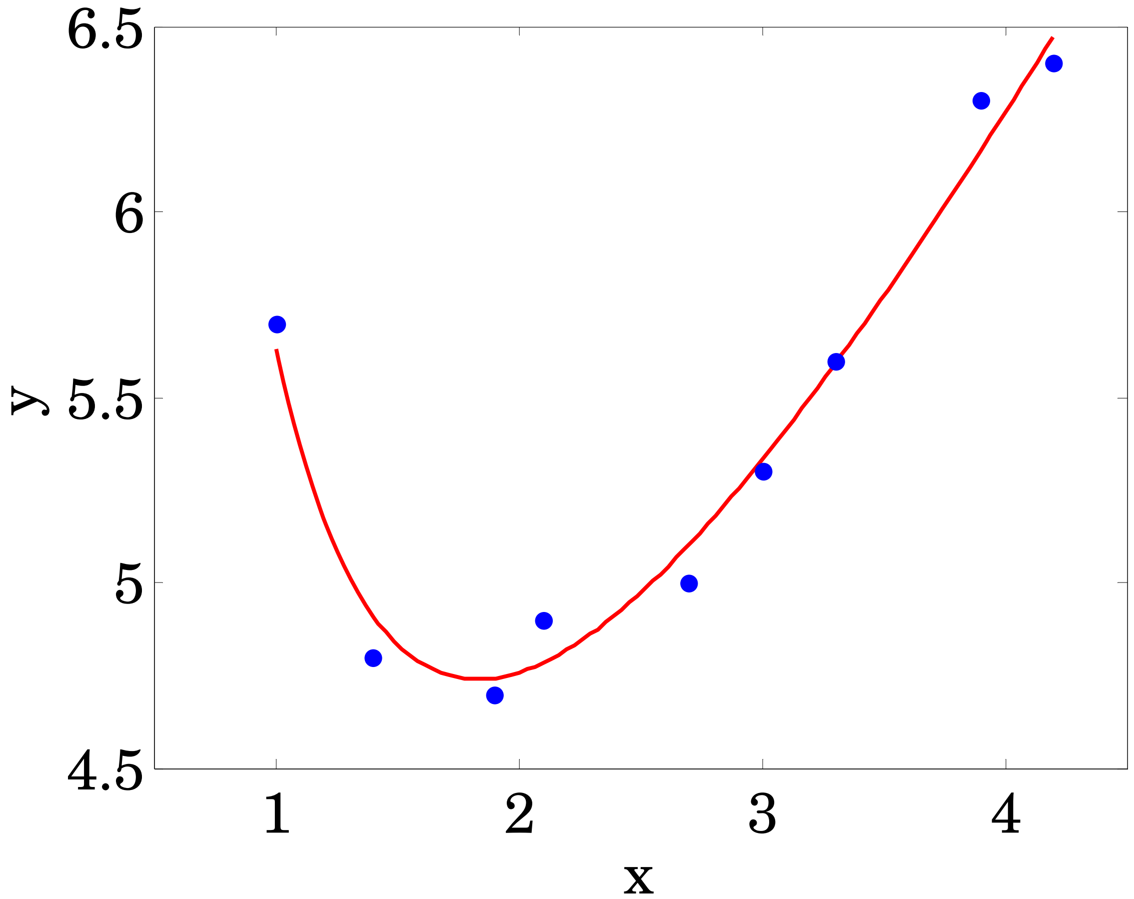
basis functions

xi	yi
1.0	5.7
1.4	4.8
1.9	4.7
2.1	4.9
2.7	5.0
3.0	5.3
3.3	5.6
3.9	6.3
4.2	6.4

```
A = [1./xi, xi];
c = A\yi
c =
    4.3350
    1.2950

x = linspace(1,4.2, 100);
y = c(1)./x + c(2)*x;

plot(x,y,'r-', xi,yi,'b.');
```

estimated model

Example 2: $y = \dfrac{c_1}{x} + c_2\, x$



Example 3: $y = \dfrac{1}{\dfrac{c_1}{x} + c_2\, x} \quad \Rightarrow \quad \dfrac{1}{y} = \dfrac{c_1}{x} + c_2\, x$

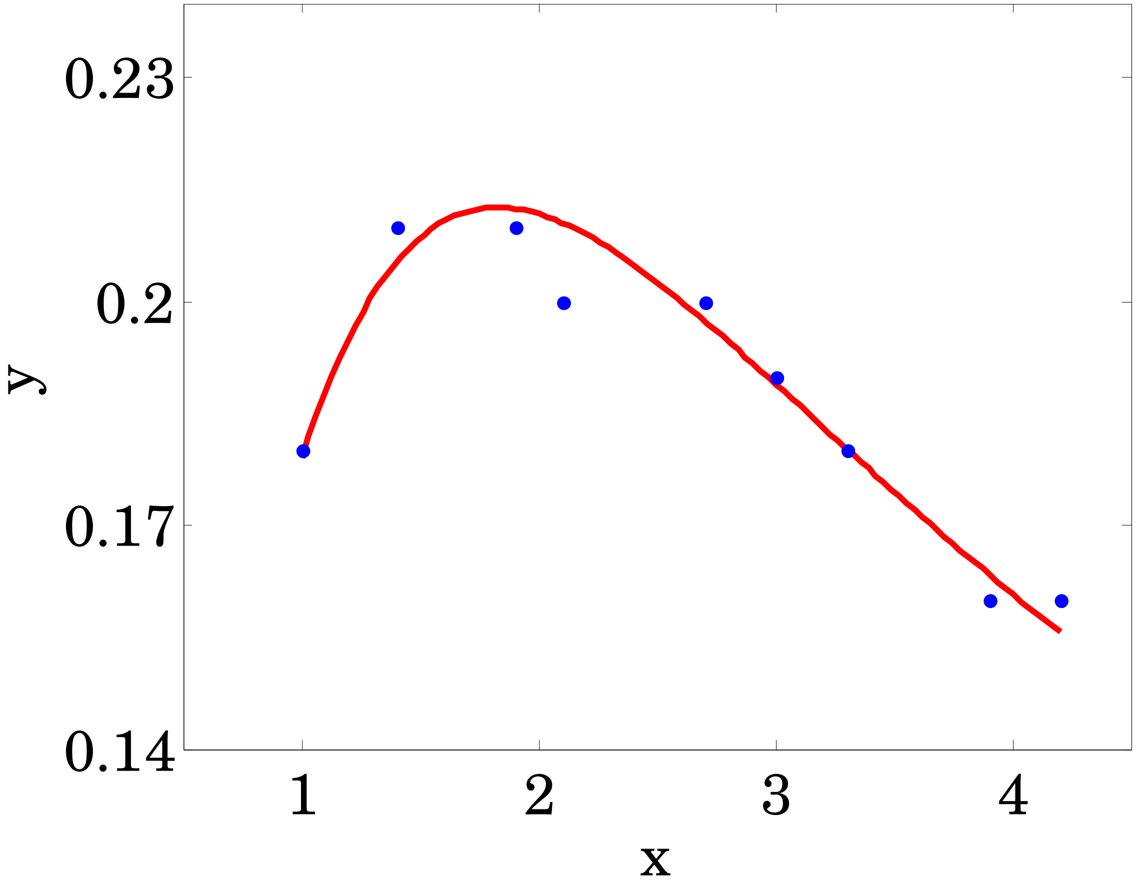
basis functions

xi	yi
1.0	0.18
1.4	0.21
1.9	0.21
2.1	0.20
2.7	0.20
3.0	0.19
3.3	0.18
3.9	0.16
4.2	0.16

```
A = [1./xi, xi];
c = A\(1./yi)
c =
    4.3012
    1.2858

x = linspace(1,4.2, 100);
y = 1./(c(1)./x + c(2)*x);

plot(x,y,'r-', xi,yi,'b.');
```

estimated model

Example 3: $y = \dfrac{1}{\dfrac{c_1}{x} + c_2\, x} \quad \Rightarrow \quad \dfrac{1}{y} = \dfrac{c_1}{x} + c_2\, x$



```
%  ti       Vi
% ------------
   0.00    2.0684
   0.05    1.6970
   0.10    1.4921
   0.15    1.2633
   0.20    1.1564
   0.25    0.9048
   0.30    0.8943
   0.35    0.6919
   0.40    0.7459
   0.45    0.5832
   0.50    0.5065
   0.55    0.4657
   0.60    0.2966
   0.65    0.3131
   0.70    0.2082
   0.75    0.2399
   0.80    0.1516
   0.85    0.0928
   0.90    0.1930
   0.95    0.2144
   1.00    0.1036
```

Example 4: $V = V_0 e^{-at}$

set-11

$$\log(V) = -a\,t + \log(V_0) \equiv p_1\,t + p_2$$

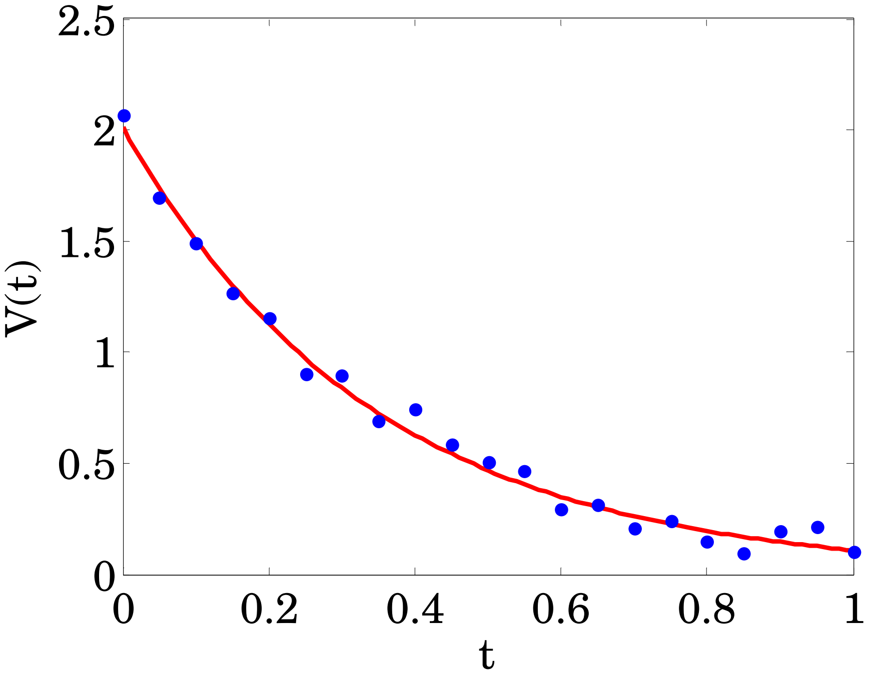
$$p = \text{polyfit}(t_i, \log(V_i), 1)$$

$$p = [p_1, p_2]$$

$$-a = p_1\,, \quad \log V_0 = p_2$$

$$\boxed{a = -p_1\,, \quad V_0 = \exp(p_2)}$$

```
A = load('capacitor.dat');
ti = A(:,1); Vi = A(:,2);
p = polyfit(ti,log(Vi),1);  % p = [-2.92,0.70];
a = -p(1), V0 = exp(p(2))
a =
    2.9170
V0 =
    2.0154
t = linspace(0,1,101);
V = exp(polyval(p,t));
% V = exp(p(1)*t+p(2));
plot(t,V,'r-',...
     ti,Vi,'b.',...
     'markersize',18);
```



Ti	ki
1	28.7
2	57.3
3	85.5
4	113
5	138
6	159
7	177
8	189
9	195
10	196
11	193
12	185
13	176
14	166
15	156
16	145
18	124
20	105
25	68
30	43
35	29
40	20.5
45	15.3

Example 5: copper data

set-11

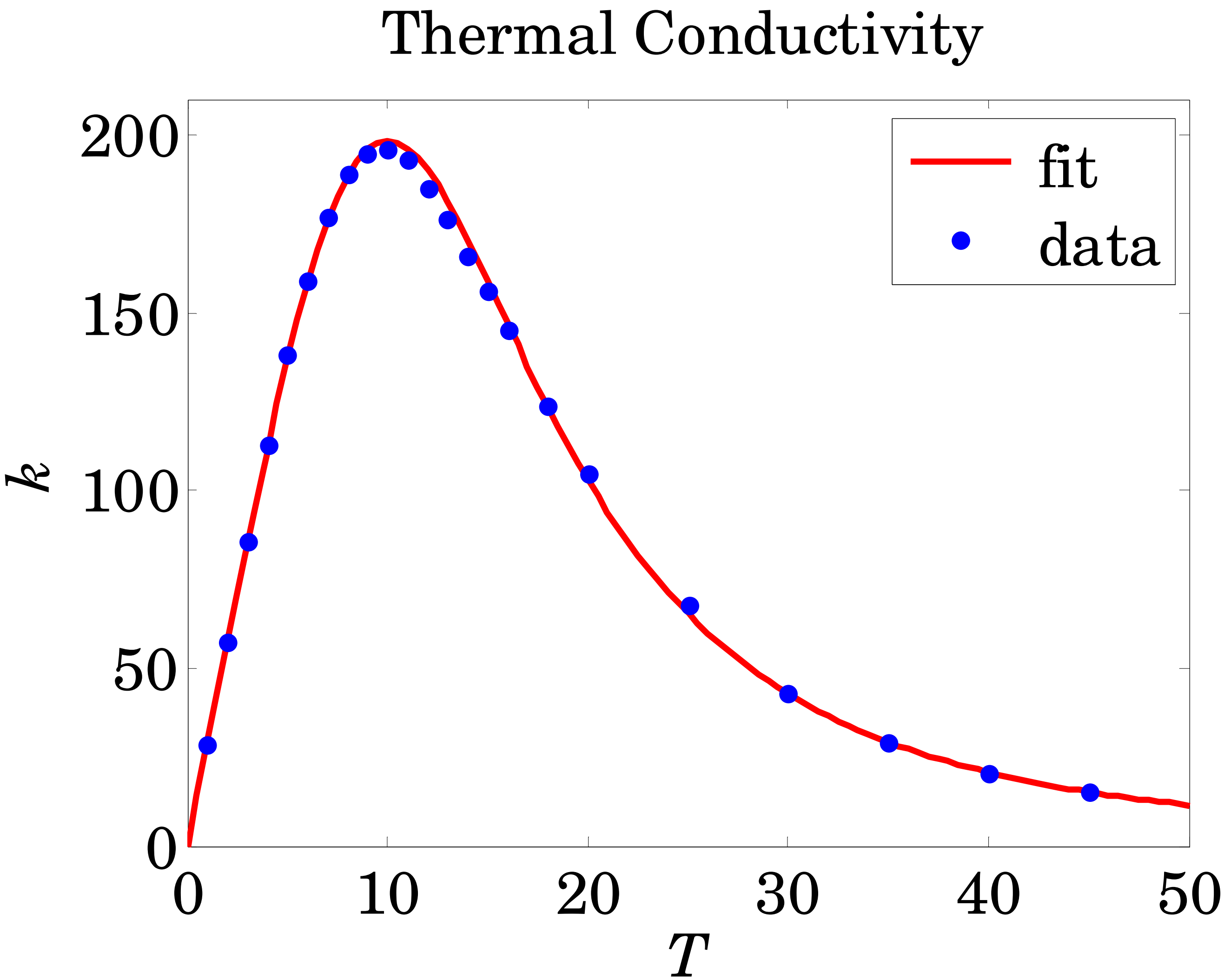
model

$$k = \frac{1}{\frac{c_0}{T} + c_1\, T + c_2\, T^2 + c_3\, T^3}$$

$$\frac{1}{k} = \frac{c_0}{T} + c_1\, T + c_2\, T^2 + c_3\, T^3$$

basis functions

```
Y = load('copper.dat');

ti = Y(:,1);
ki = Y(:,2);
yi = 1./ki;

A = [1./ti, ti, ti.^2, ti.^3];    % basis

c = A\yi;    % fit 1/k to model

T = linspace(0,50,101);

k = T./(c(1) + c(2)*T.^2 + c(3)*T.^3 + ...
        c(4)*T.^4);

plot(T,k,'r-', ti,ki,'b.','markersize',18);
```



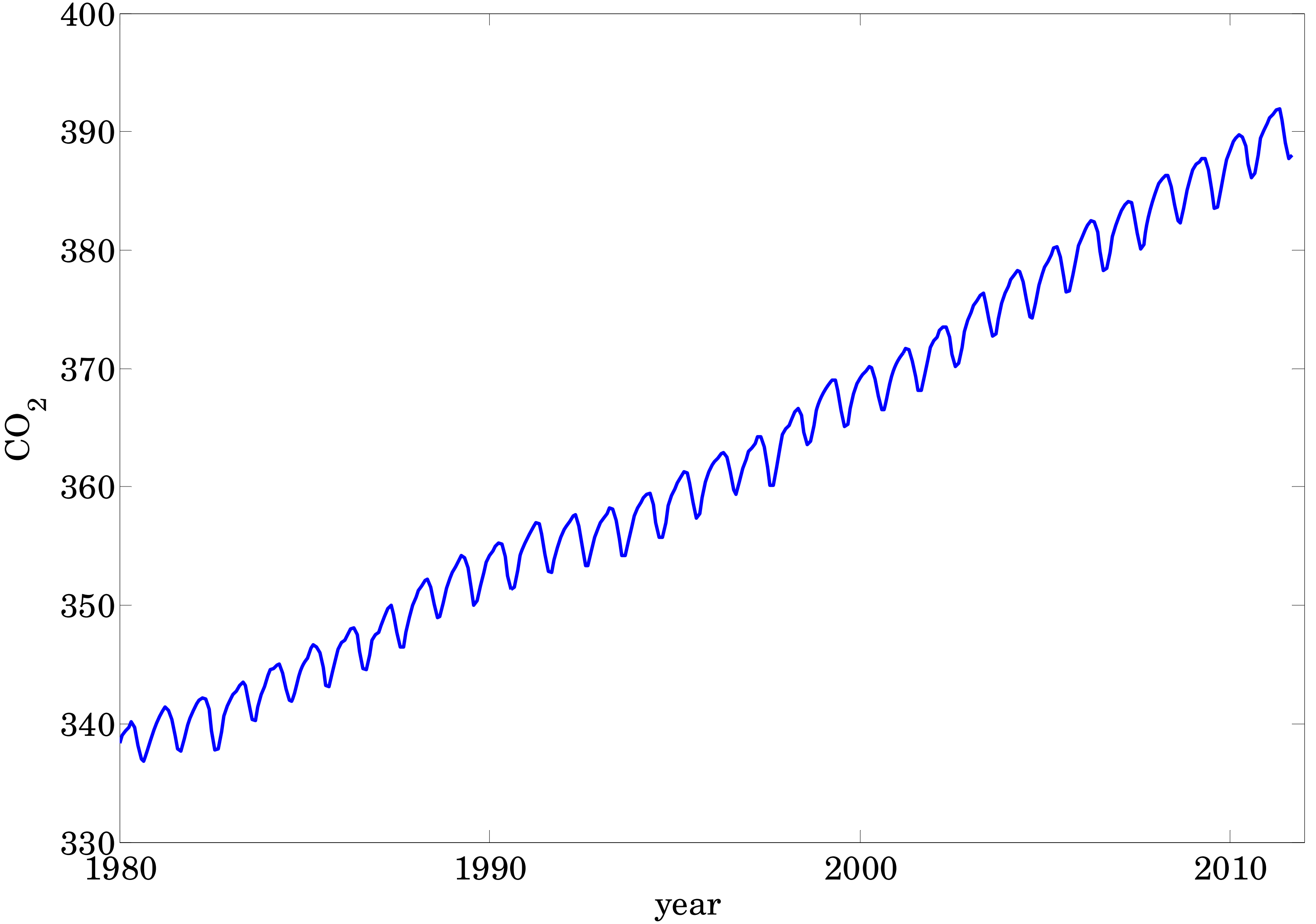
Thermal Conductivity
fit
data
k
T
200
150
100
50
0
0
10
20
30
40
50

Example 6: CO2 emissions

```
% year   month   decimal      average     trend
% -----------------------------------------------
  1980     1    1980.042      338.33      337.70
  1980     2    1980.125      339.04      337.99
  1980     3    1980.208      339.36      338.02
  1980     4    1980.292      339.74      338.12
  1980     5    1980.375      340.16      338.64
  1980     6    1980.458      339.73      338.94
  1980     7    1980.542      338.20      339.05
  1980     8    1980.625      336.99      339.24
  1980     9    1980.708      336.81      339.20
  1980    10    1980.792      337.57      338.91
  1980    11    1980.875      338.69      339.03
  1980    12    1980.958      339.41      339.19
  ...    ...      ...           ...         ...
  2011     7    2011.542      389.04      390.27
  2011     8    2011.625      387.76      390.48
  2011     9    2011.708      388.04      390.80
```

file: co2.dat on sakai

ftp://ftp.cmdl.noaa.gov/ccg/co2/trends/co2_mm_gl.txt


400
390
380
370
360
350
340
330
CO_2
1980
1990
2000
2010
year

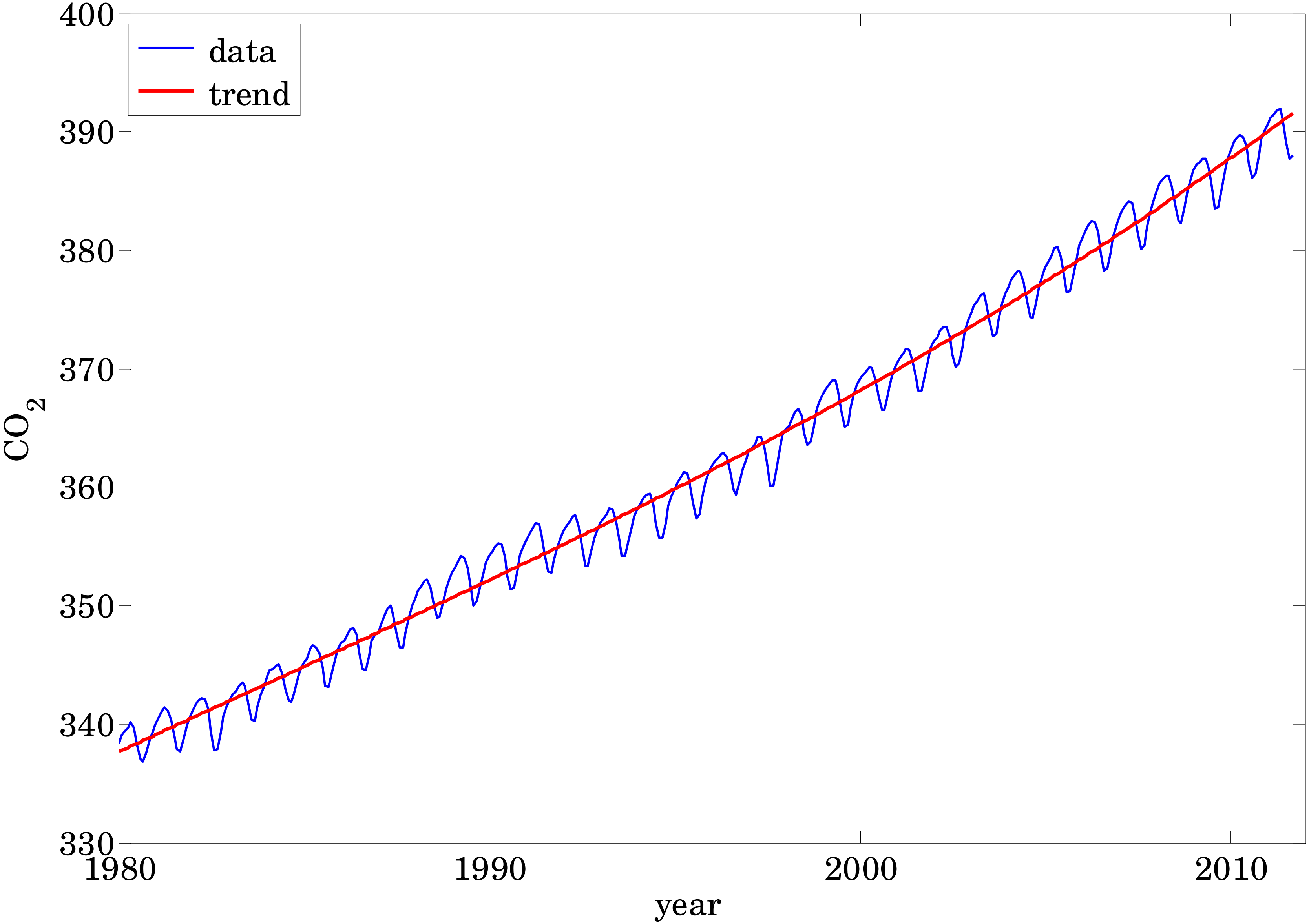
```
Y = load('co2.dat');            % file on sakai

y = Y(:,4);
t = (0:length(y)-1)';
ty = t/12 + 1980;               % rescale time

figure; plot(ty, y, 'b-');
xaxis(1980, 2012, 1980:10:2010);
xlabel('year     '); ylabel('CO_2');
```

model

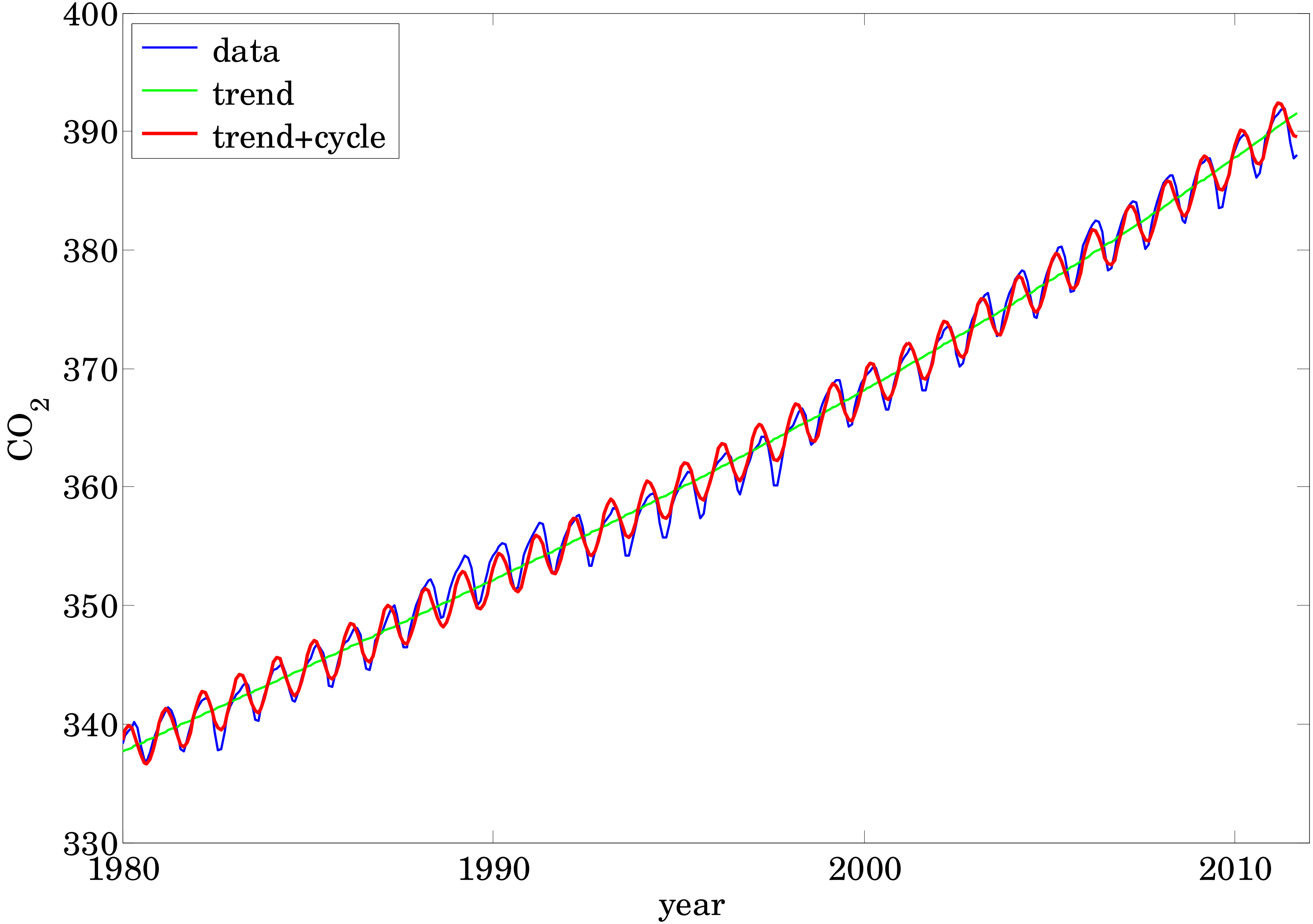
$$y = \underbrace{c_0 + c_1 t + c_2 t^2 + c_3 t^3}_{\text{trend component}} + \overbrace{c_4 \cos\left(\frac{2\pi t}{12}\right) + c_5 \sin\left(\frac{2\pi t}{12}\right)}^{\text{cyclical component}}$$

basis functions

```
c = [t.^0, t, t.^2, t.^3, ...
     cos(2*pi*t/12), sin(2*pi*t/12)] \ y;

T = @(t) c(1) + c(2)*t + c(3)*t.^2 + ...
         c(4)*t.^3;                          % trend

C = @(t) c(5)*cos(2*pi*t/12) + ...
         c(6)*sin(2*pi*t/12);                % cycle

figure; plot(ty, y, 'b-', ty,T(t),'g-', ...
             ty,T(t)+C(t),'r-');

xaxis(1980, 2012, 1980:10:2010);
xlabel('year     '); ylabel('CO_2')
legend(' data', ' trend', ' trend+cycle',...
       'location','nw');
```



400
390
380
370
360
350
340
330
1980
1990
2000
2010
year
CO_2
data
trend



data
trend
trend+cycle
400
390
380
370
360
350
340
330
1980
1990
2000
2010
year
CO_2