Use latex for this assignment! Submit a pdf file to the dropbox.

You do not need to call it `am584hw3.pdf`, but please submit a single `pdf` file.

Problem #1.

Consider the data set (t_i, y_i) consisting of the following 4 points:

$$(0, 2.1), \quad (1, 2.5), \quad (4, 3), \quad (9, 3.6).$$

Suppose we want to compute a least squares fit of the form $f(t) = c_0 + c_1\sqrt{t}$.

(a) Set up the desired matrix equation $Ax = b$ where $x \in \mathbb{R}^2$ and $b \in \mathbb{R}^4$. What are A and b and what do the components of x represent?

You can use Matlab/Python for the rest of the problem and paste your commands and the results into the latex file. Use the `verbatim` environment as illustrated in the latex file for this assignment.

(b) The least squares solution x can be computed using the backslash operator in Matlab or the `lstsq` command in Python (assuming you start IPython with the `-pylab` flag).

Solve the least squares problem. Then plot the resulting function $f(t)$ over the interval $0 \leq t \leq 10$ along with the data points. Do this by evaluating $f(t)$ at 1000 points in the interval.

See the examples on the course webpage for how to do such plots in Matlab or Python: `http://faculty.washington.edu/rjl/amath584aut2011/lstsq.html`

(c) Compute the QR factorization of A and print it out.

Then solve the least squares problem by solving the linear system $Rx = Q^T y$. (This can be solved with the backslash in Matlab or with the `solve` function in Python).

Confirm that you get the same coefficients as in part (b).

(d) Project the vector b onto the column space of A to get $Pb \in \mathbb{R}^4$. Plot the ith element of Pb vs. t_i (for $i = 1, 2, 3, 4$) on the same plot as above. Explain why these points lie where they do.

(e) Another way to solve the least squares problem is to form the normal equations $A^T Ax = A^T b$. compute this matrix and right hand side and solve the system. Confirm that this gives the same coefficients as before.

(f) The pseudo-inverse $A^+ \in \mathbb{R}^{2 \times 4}$ can be computed either as $A^+ = (A^T A)^{-1} A^T$ or as $A^+ = R^{-1} Q^T$. Compute A^+ both ways and confirm that you get the same result.

(g) Compute $A^+ A$ and AA^+. One of these should be an identity matrix. Make sure you understand why one is and the other isn't (no need to say anything about it in your solutions).

Solution:

Problem #2.

Download the file `http://faculty.washington.edu/rjl/amath584aut2011/_static/co2.txt`, which contains a set of data that you will need for this assignment. You should be able to click on the link above.

The resulting file `co2.txt` contains 90 lines of comments followed by 636 lines of data. The third column is time (in years) and the fourth column is CO_2 concentration in the atmosphere (in parts per million) as measured at Mauna Loa. These give a set of (t_i, y_i) values that we will apply least squares to. See `http://www.esrl.noaa.gov/gmd/ccgg/trends/mlo.html` for more information on this data.

Plotting this data should give:
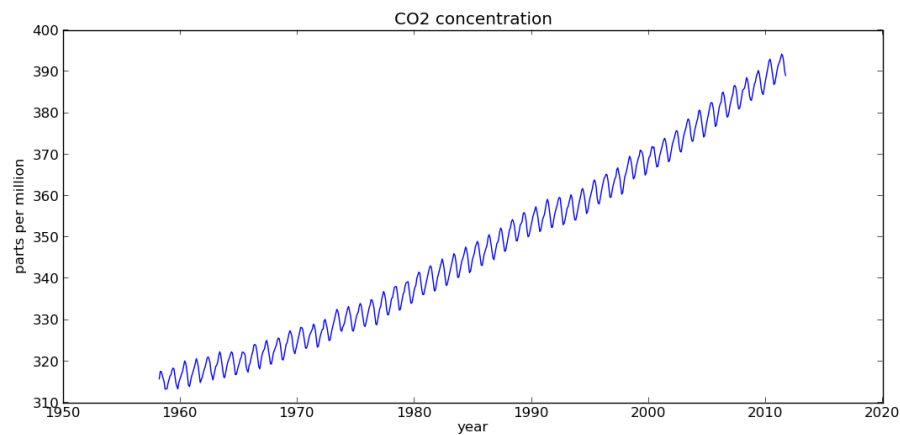
To produce such a plot (and to do the rest of this assignment) you will have to read the data into Matlab or Python and first pre-process it by stripping out any (t, y) values for which $y = -99.99$ (indicating that data was not available at this time).

The codes on the next page show how to do this, and print out the figure as a `png` file that can be incorporated in your latex as illustrated in this file.

Matlab:

```
% Read column data with whitespace delimiter, skipping 90 rows:
d = importdata('co2.txt',' ',90)
data = d.data;
t = data(:,3);
y = data(:,4);

% filter out the bad data:
good = y>0;    % indices of good data
t = t(good);   % this portion of array
y = y(good);

plot(t,y)
title('CO2 concentration')
xlabel('year')
ylabel('parts per million')
print -dpng co2figure.png
```

Python:

```
from pylab import *

# Read column data with whitespace delimiter, skipping 90 rows:
data = loadtxt('CO2.txt',skiprows=90)
t = data[:,2]   # First column is index 0
y = data[:,3]

# filter out the bad data:
good = y>0   # indices of good data
t = t[good]  # this portion of array
y = y[good]

figure(1)
clf()
plot(t,y)
title('CO2 concentration')
xlabel('year')
ylabel('parts per million')
savefig('co2figure.png')
```

There is a clear oscillation in this data with a period of one year, due to seasonal variations. There is also an upward trend that might be fit with a linear function as the first attempt. This suggests we perform a least squares fit with the basis functions

$$\phi_0(t) = \cos(2\pi t), \quad \phi_1(t) = \sin(2\pi t), \quad \phi_2(t) = 1, \quad \phi_3(t) = t - 1985.$$

The shift by 1985 in the last basis function keeps the values in the matrix smaller, and becomes particularly important when we add higher powers.

(a) Use Matlab or Python to compute the least squares fit. Plot the fit as a red curve on the same plot.
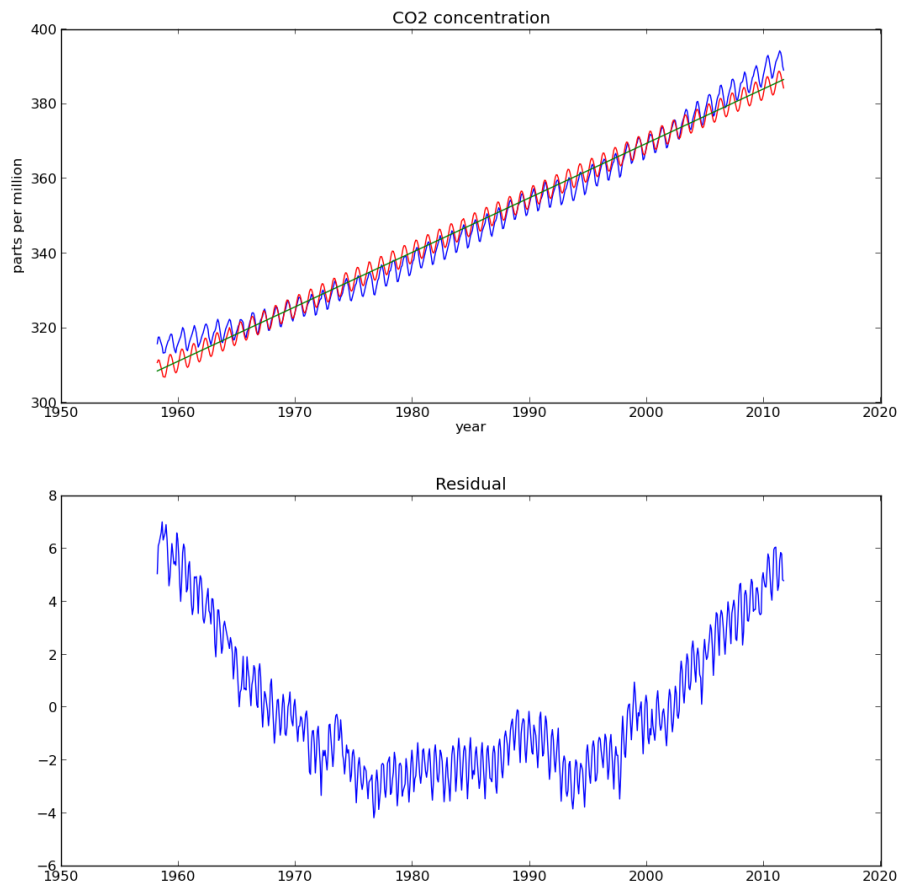
To see the trend without the oscillation, take the coefficients $c \in \mathbb{R}^4$ just found and plot the function

$$p(t) = c_2\phi_2(t) + c_3\phi_3(t)$$

on the same plot, as a green curve.

Also plot the residual as a function of time on a separate plot.

You should get something like these:

4

(b) This fit doesn't match the trend very well, so we might try improving it by using a quadratic function instead of linear for the "secular terms", especially since the residual looks something like a parabola. This can be done by adding another basis function $\phi_4(t) = (t - 1985)^2$. Repeat part (a) using this. Plot the fit, the quadratic trend, and the residual.

(c) This fit looks good enough that we might be tempted to use it to predict the future. Evaluate the function $f(t)$ over the interval $1960 \leq t \leq 2050$ with enough points to show how it behaves and plot this.

As a sanity check we might also try extrapolating backwards in time. Plot the same function $f(t)$ over the interval $1800 \leq t \leq 2010$ and comment on what you see.

Solution: