12.540 Homework #2 Due Monday, April 18, 2012

Matlab solution codes are given in HW02_2012.m This code uses cells and contains the solutions to all the questions.

Question 1: *Non-linear estimation problem*
The data below are from a model of data of the form:
*y(t) = Acos(2*pi*t/15+phi) + Y0 + N(t)*
where A, the amplitude, phi the phase, and Y0 are unknown parameters to be estimated. N(t) is Gaussian noise with standard deviations given by the +- column (all constant in this case).
Data:

Time	y(t)	+-
0.0	14.02	1.00
1.0	9.60	1.00
2.0	2.98	1.00
3.0	0.61	1.00
4.0	-3.68	1.00
5.0	-7.19	1.00
6.0	-10.64	1.00
7.0	-9.83	1.00
8.0	-7.26	1.00
9.0	-5.24	1.00
10.0	2.38	1.00
11.0	6.14	1.00
12.0	9.68	1.00
13.0	12.23	1.00
14.0	14.27	1.00
15.0	12.51	1.00
16.0	9.28	1.00
17.0	5.23	1.00
18.0	1.93	1.00
19.0	-3.30	1.00
20.0	-6.41	1.00
21.0	-7.43	1.00
22.0	-9.58	1.00
23.0	-6.59	1.00
24.0	-3.76	1.00
25.0	2.61	1.00
26.0	7.37	1.00
27.0	10.79	1.00
28.0	11.75	1.00
29.0	13.82	1.00

(a) Estimate A, phi, and Y0 using a non-linear weighted least-squares estimator and give the values and their standard deviations.

(b) Reformulate the problem to be of the form:
*y(t) = Ac*cos(2*pi*t/15) + As*sin(2*pi*t/15) + Y0 + N(t)*
and estimate Ac, As, and Y0 using linear weighted least-squares estimator and give the estimates and their standard deviations. The Ac and As values are quadrature components.
(c) Use the Ac and As values and their variance-covariance matrix from part (b) to compute the amplitude and phase and their variance-covariance matrix. Compare the estimated with those from part (a).
(d) These data are uniformly spaced in time and constant standard deviations. Determine the quadrature components from an FFT of the data. Compare the results with part (b). Why are the results the same?

Solution

(1a): This is a standard non-linear least squares problem that can be solved by Taylor series expansion. Using the non-linear equation we can write:

$$(1.1)\quad y(t) = A\cos(2\pi t/15 + \phi) + Y0 + N(t)$$

$$(1.2)\quad y(t) = (A_0 + \delta A)\cos(2\pi t/15 + \phi_0 + \delta\phi) + (Y0 + \delta Y0) + N(t)$$

$$(1.3)\quad y_0(t) = A_0\cos(2\pi t/15 + \phi_0) + Y0_0$$

$$(1.4)\quad \delta y(t) = \cos(2\pi t/15 + \phi_0)\delta A - A_0\sin(2\pi t/15 + \phi_0)\delta\phi + \delta Y0$$

$$(1.5)\quad y(t) - y_0(t) = \delta y(t)$$

So initial values are chosen for A_0, ϕ_0 and $Y0_0$, the least squares solution uses the difference between the observed values and the values computed with the initial values (1.5), and the least squares uses the partial derivatives (or Jacobian) given in (1.4). Estimates of δA, $\delta\phi$ and $\delta Y0$ are returned by the least squares solution. These values are added to the initial values and the process iterated until the estimates of δA, $\delta\phi$ and $\delta Y0$ are small compared to their standard deviations.
When initial values of A_0=1, ϕ_0=0, and $Y0_0$=0 are used, the iteration from the homework solution is given below. The Error entry is the sum of the ratio, squared, of the adjustments at each iteration to the standard deviations of the estimates. The iteration is continued until this sum is less than 10^{-4}.

```
Q 1(a): Iteration   1 Error 1.854e+03
Ampl       10.41 dA  9.40965 +-     0.26
Phase     5.0321 dP  5.03215 +-   0.2582
Offset      2.21 dO  2.20967 +-     0.18

Q 1(a): Iteration   2 Error 4.101e+03
Ampl       -1.50 dA -11.91456 +-     0.26
Phase     6.1334 dP  1.10127 +-   0.0248
Offset      2.21 dO  0.00000 +-     0.18

Q 1(a): Iteration   3 Error 2.470e+03
Ampl        9.54 dA 11.04718 +-     0.26
```

```
Phase    1.7949 dP -4.33850 +-   0.1716
Offset     2.21 dO -0.00000 +-     0.18

Q 1(a): Iteration   4 Error 2.629e+03
Ampl       2.59 dA -6.94952 +-     0.26
Phase    0.6141 dP -1.18082 +-   0.0271
Offset     2.21 dO  0.00000 +-     0.18

Q 1(a): Iteration   5 Error 1.219e+03
Ampl      11.41 dA  8.81462 +-     0.26
Phase   -0.1131 dP -0.72723 +-   0.0996
Offset     2.21 dO -0.00000 +-     0.18

Q 1(a): Iteration   6 Error 6.120e+02
Ampl       9.78 dA -1.63235 +-     0.26
Phase    0.4282 dP  0.54133 +-   0.0226
Offset     2.21 dO -0.00000 +-     0.18

Q 1(a): Iteration   7 Error 4.874e+01
Ampl      11.56 dA  1.78431 +-     0.26
Phase    0.4543 dP  0.02613 +-   0.0264
Offset     2.21 dO  0.00000 +-     0.18

Q 1(a): Iteration   8 Error 3.279e-02
Ampl      11.56 dA  0.00273 +-     0.26
Phase    0.4503 dP -0.00404 +-   0.0223
Offset     2.21 dO  0.00000 +-     0.18

Q 1(a): Iteration   9 Error 1.351e-07
Ampl      11.56 dA  0.00009 +-     0.26
Phase    0.4503 dP  0.00000 +-   0.0223
Offset     2.21 dO -0.00000 +-     0.18
```

Notice here that the initial iterations are quite unstable because of rapid changes in the phase. Trying different initial values for A can change this behavior dramatically. When angles are part of a non-linear inversion, the initial amplitudes should be approximately correct. Also notice that the Y0 term is a linear term and it converges within the first iteration. This behavior is common although, I don't know if it universal. (In this case, the non-linear problem can be formulated as a linear one (Q1b) and this might explain this behavior.

Although not asked, we can compute the coefficients from Q1b and the results are shown below.

```
Cosine     10.41 +-      0.26
Sine       -5.03 +-      0.26
```

(1.b) The linear estimation problem is solved by cosine into a cosine and sine component using the cosine sum rule.

$$A\cos(2\pi t/15+\phi) = A\cos(\phi)\cos(2\pi t/15) - A\sin(\phi)\sin(2\pi t/15)$$

We now estimate Ac=Acos(ϕ) and As=Asin(ϕ) and the problem is linear in Ac and As. The results of this estimation (in one iteration because the problem is linear) are:

```
Q 1(b): Linear estimates
Offset     2.21 +-     0.18
Cosine    10.41 +-     0.26
Sine      -5.03 +-     0.26
```

(1.c) These estimates can be transformed to amplitude and phase using the non-linear equations for the estimates and the linearized equations for propagating the variance-covariance matrix

$$A = \sqrt{Ac^2 + As^2}$$

$$\phi = \tan^{-1}(-As/Ac)$$

$$\begin{bmatrix} dA \\ d\phi \end{bmatrix} = \begin{bmatrix} Ac/A & As/A \\ -As/A^2 & Ac/A^2 \end{bmatrix} \begin{bmatrix} dAc \\ dAs \end{bmatrix}$$

Substituting the results and using propagation of covariances yields results exactly the same as those obtained from the non-linear estimates.

```
Q 1(c): Linear estimates transformed to amplitude and phase
Amplitude     11.56 +-     0.26
Phase        0.4503 +-   0.0223 rad
```

(1.d) The Matlab FFT routine can be used and based on the normalization used by this FFT, the results can be directly extracted as the 3rd elements in the FFT. Care is needed depending on the FFT routine as to whether the spectrum is wrapped around zero frequency and the normalization used for the number of data in the FFT.

```
Q 1(d): FFT Estimates, Linear and Non-linear estimates
Cosine: FFT  10.409649 Lin 10.409649 NonL 10.409649
Sine  : FFT  -5.032150 Lin -5.032150 NonL -5.032150
```

The results agree with the least squares estimates because the length of data and frequency are commensurate and this the frequency is one of the Fourier frequencies. These frequencies are orthogonal and thus when used in an estimation (with constant data variances), the normal equations are diagonal and thus the problem can be solved one frequency at a time. The results are the same as shown above.

Question 2: *Linear estimation problem with regular and sequential estimation. Using the data set listed below:*
(a) Fit a linear regression (y=ax+b) to the data using standard weighted least squares
(b) Compute the chi-squared per degree of freedom of the postfit residuals. Is

this value consistent with the variations expected from random variations?
(c) Divide the data into two parts (first and last 5 data points) and use sequential estimation to determine a and b.
(d) Find the weighted mean y (the weighted mean is the weighted least squares solution to the equation y = b). Compare this estimate of b with the estimates of the a' when x is shifted such that estimates of a' and b' are uncorrelated (i.e., the solution to y=a'(x-xo)+b' where xo is selected such that the estimates of a' and b' are uncorrelated.*

Data:

X	Y	+-
1	5.759	1.050
2	-2.389	1.080
3	-10.512	0.975
4	-18.220	0.974
5	-19.745	1.075
6	-24.429	1.227
7	-32.521	1.025
8	-36.411	0.817
9	-43.655	0.989
10	-50.882	0.842

Solution:

(2.a) Straightforward problem using the standard weighted least squares solution. Output from Matlab is:

```
Q 2 (a) : Estimates:
Offset   9.01 +-   0.69
Rate    -5.89 +-   0.11
```

(**2.b**) There are two ways to solve this problem. The most straightforward is to compute the postfit residuals and directly compute χ^2. The other more elegant way to compute the so-called prefit χ^2 (This is the χ^2 calculation using the original non-fitted data) and then compute the change to from the parameter estimates. This change is simply the solution vector dotted with $\mathbf{A}^T\mathbf{W}\mathbf{y}$ where **y** is the original data. Both calculations are shown below and generate identical results. The advantage of the latter formulation is that the postfit residuals themselves never need to be computed which can be a major advantage for large data sets.

```
Q 2 (b): Chi**2/f
Chi**2/f from postfit residuals     4.1221 for 8 dof
Chi**2/f from prefit  residuals     4.1221 for 8 dof
```

The probability that could be this large due to random error can be computed from the chi-squared distribution (see http://mathworld.wolfram.com/Chi-SquaredDistribution.html).

(Be careful with the gamma function definition differences between this site and Matlab; you need to verify the integral bounds being used in the incomplete expressions).

```
Probability that chi**2 would be greater than this due to
random error:   0.0062 %
```

There is a small probability that the noise in the data is consistent with the error bars (in fact the true noise is 50% larger than the errors (see matlab code).

(2.c) This is a straightforward application of sequential estimation. The important step here is to use the covariance matrix of the two sets of estimates when using them the obtaining the combined results. Results from the matlab code are given below. The small differences from 1.a are due to rounding error.

```
Q 2 (c) : Estimates for data 1-5:
Offset  11.08 +-   1.11
Rate    -6.75 +-   0.33
Q 2 (c) : Estimates for data 6-10:
Offset  13.99 +-   2.67
Rate    -6.44 +-   0.32
```

We then use the two pairs of estimates above with the full covariance matrices from the least squares solution to estimate the offset and rate parameters. In this case the partials matrix is a pair of unit matrices because the parameters and data are the same quantities.

```
Q 2 (c) : Estimates from combined data:
Offset   9.01 +-   0.69
Rate    -5.89 +-   0.11
Differences from 1(a): -1.7053e-13 4.6185e-14
```

(2.d) A method of making the parameter estimates uncorrelated is to modify the time argument for the offset estimate. In the Matlab code this method is used to determine the time the offset is referred to (ie., a Dt shift so that the offset refers to a time at the “center” of the data. These estimates are given below.

```
Q 2 (d) : Estimates of uncorrelated parameters by shifting
time     5.8602:
Offset-shift -25.5087 +- 0.3119
Rate-shift   -5.8903 +- 0.1058
Covariance  1.9326e-18
```

Compute the weighted mean. The solution code shows the method. This is a least squares solution with the observation equation y=x. Notice that the weighed mean matches the offset estimate at the “uncorrelated time”.

```
Q 2 (d): Weighted Mean   -25.5087 +-     0.3119
Compare with Offset-shift above
```

Question 3: *Example of using differenced data*
Using the data below:
*(a) Estimate A given the model that y(t) = A*sin(X)+C(t) where C(t) is a randomly varying function of time.*
(b) Instead of estimating C(t) at each time, use a differencing method to eliminate

C(t) from the estimation. (Hint: Use propagation of variances to determine the covariance matrix of the differenced data.
(c) Determine A by treating C(t) as correlated data noise (i.e., the C(t) noise is that same at each time and therefore highly correlated at each time but independent between times).
(d) Compare the results from the "brute force", differencing, and data noise approaches (Hint: When solved correctly they all generate the same result).

Data:

T	X	Y	+-
1	1.1000	44.1946	1.00
1	0.3000	36.8838	1.00
1	-0.1900	32.0616	1.00
2	1.2000	-0.1419	1.00
2	0.1000	-5.9336	1.00
2	-0.1600	-9.1461	1.00
3	1.3000	-7.8164	1.00
3	-0.1000	-19.3260	1.00
3	-0.1100	-21.5875	1.00
4	1.4000	35.3201	1.00
4	-0.3000	24.3602	1.00
4	-0.0400	26.2854	1.00
5	1.5000	-34.5531	1.00
5	-0.5000	-47.7803	1.00
5	0.0500	-41.8605	1.00
6	1.6000	0.6797	1.00
6	-0.7000	-13.2281	1.00
6	0.1600	-6.1792	1.00
7	1.7000	1.1492	1.00
7	-0.9000	-17.6524	1.00
7	0.2900	-8.0827	1.00
8	1.8000	25.4898	1.00
8	-1.1000	6.2613	1.00
8	0.4400	16.9865	1.00
9	1.9000	16.7815	1.00
9	-1.3000	-4.4243	1.00
9	0.6100	11.4170	1.00
10	2.0000	-14.0549	1.00
10	-1.5000	-37.0554	1.00
10	0.8000	-14.9231	1.00

Solution

(3.a) The brute estimates are straightforward and yields. The postfit chi-squared is also computed and shown below.

```
Q 3 (a) Brute force estimates
X value Estimate    10.2884 +-     0.2765 Error     0.2884
Clock offsets  1    34.2912 +-     0.5846 Error     0.3061
Clock offsets  2    -8.0663 +-     0.5829 Error     0.6161
Clock offsets  3   -18.8289 +-     0.5815 Error     0.7904
Clock offsets  4    26.4263 +-     0.5805 Error    -1.0007
```

```
Clock offsets  5   -43.3461 +-     0.5797 Error    -0.5993
Clock offsets  6    -8.0076 +-     0.5793 Error    -0.2432
Clock offsets  7    -9.8904 +-     0.5791 Error     0.9086
Clock offsets  8    14.5017 +-     0.5793 Error     0.0186
Clock offsets  9     6.0193 +-     0.5796 Error    -0.4000
Clock offsets 10   -24.1688 +-     0.5803 Error    -1.1223
Chi^2 per degree of freedom 2.0926; f =  19, Probability     0.35 %
```

(3.b) For the difference solution, a difference operator is created that differences the data between epochs 1-2 and 1-3. In this solution, the complete difference operator is created but it could have been for each measurement time separately (and sequential). This sequential approach would be method used for a large data set. The important step here, which can again be done sequentially, is to form the covariance matrix of the differenced observations. Using the differenced observations to estimate just the one parameter yields and again with postfit chi-squared, exactly the same result.

```
Q 3 (b) Differenced data result
Solution from differnced data :    10.2884 +-     0.2765
Difference from full solution : 0.0000e+00 +- 0.0000e+00
Chi^2 per degree of freedom 2.0926; f =  19, Probability     0.35 %
```

As an example below, I also include the results if the covariance matrix in not used. The result is similar, but certainly not the same. Notice also that the sigma estimate is optimistic.

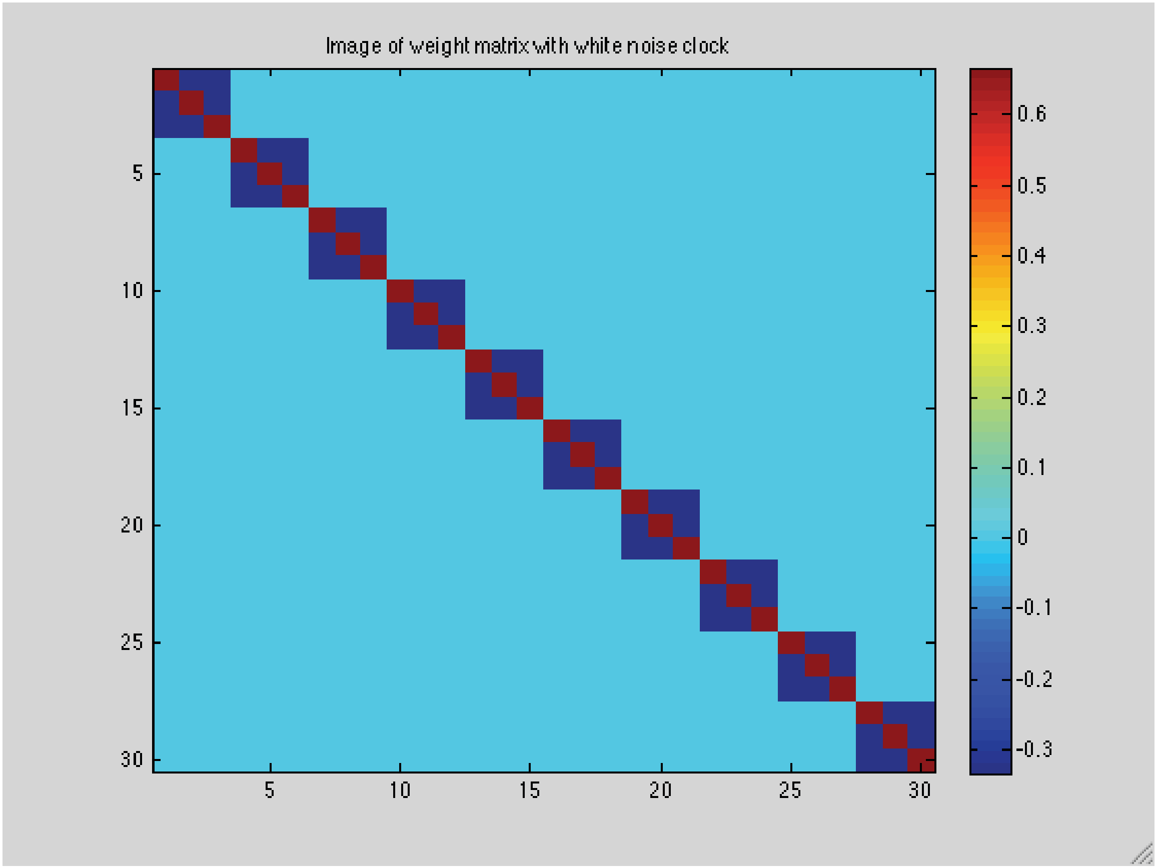
```
Q 3 (b) Differenced data result with no-correlations
Solution from differenced data:    10.2321 +-     0.1833
Difference from full solution : -5.6268e-02 +- -9.3220e-02
```

When the differenced data covariance matrix is used, the results are identical. In more complicated geometries this result holds if all the non-dependent differences are found. In some geometries other combinations are needed in addition to simple differences are needed to represent the data.

(3.c) We can also solve this problem but estimating just the A value, using the complete data and by treating the clock term as noise. We add a correlated contribution to the data covariance matrix and then solve for just A. In the solution we use 10000 as the standard deviation of the clock "noise" and the results generated are the same as the other two solution. The chi^2 per degree of freedom is also the same when 19 degrees of the freedom are used. (There are only 20 "non-zero" eigenvalues in the weight matrix minus the 1 parameter that was estimated resulting in 19 degrees of freedom.).

```
Q 3 (c) White noise model result
Solution from white noise     :    10.2884 +-     0.2765
Difference from full solution : -1.6023e-09 +- 5.3453e-10
Chi^2 per degree of freedom 2.0926; f =  19, Probability     0.35 %
Chi^2 from prefit  39.7590 and from postfit residuals  39.7590
```

The weight matrix from the white-noise clock solution is shown below revealing its rather simple structure.



(3.d) All results, including the statistical parameters, are exactly the same for all three analysis methods used. Notice in this also, that although the data noise is consistent with the data variances, the chi^2/f probability is only 0.35%. Changing the seed at the top of the program will change this to more typical high probability values.