Homework 4 solution

Due date: March 9, 2016, 11:55PM.

Bibliography: Course lecture notes Lessons 15-17. Textbook pp. 189-216, 256-267, Sections 4.2-4.4, 5.5.

Feel free to use Octave within the theoretical exercises to avoid tedious arithmetic once you've become familiar with basic procedure.

Highlights of solution procedures.

1. (1 course point) Textbook p.195, Exercise 4.3.15

   Solution. The two approaches to solution of least squares problems, $\min_{\boldsymbol{x}\in\mathbb{R}^n}\|\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}\|$, $\boldsymbol{x}\in\mathbb{R}^n$, $\boldsymbol{b}\in\mathbb{R}^m$, $\boldsymbol{A}\in\mathbb{R}^{m\times n}$, $m>n$, are:

   i. Solve normal equations: $(\boldsymbol{A}^T\boldsymbol{A})\boldsymbol{x}=\boldsymbol{A}^T\boldsymbol{b}$, suitable for small values of n, but generally unstable to small numerical errors

   ii. Projection on $C(\boldsymbol{A})$: (a) Compute factorization $\boldsymbol{Q}\boldsymbol{R}=\boldsymbol{A}$, $\boldsymbol{Q}\in\mathbb{R}^{m\times n}$, $\boldsymbol{R}\in\mathbb{R}^{n\times n}$; (b) solve triangular system $\boldsymbol{R}\boldsymbol{x}=\boldsymbol{Q}^T\boldsymbol{b}=\boldsymbol{c}$.

   (a) $m=3, n=1$. By the normal equation method

$$\boldsymbol{A}^T\boldsymbol{A}=6, \boldsymbol{A}^T\boldsymbol{b}=3\Rightarrow x=\frac{1}{2}$$

The arithmetic can be carried out in Octave.

```
octave> A=[1; 2; -1]; N=A'*A; disp(N);
   6
octave> b=[1; 1; 0]; c=A'*b; disp(c);
   3
octave>
```

The solution can also be obtained by the projection method

$$\boldsymbol{A}=\left(\begin{array}{c}1\\2\\-1\end{array}\right)=\left(\begin{array}{c}\frac{1}{\sqrt{6}}\\\frac{2}{\sqrt{6}}\\-\frac{1}{\sqrt{6}}\end{array}\right)\sqrt{6}, c=\boldsymbol{Q}^T\boldsymbol{b}=\left(\begin{array}{ccc}\frac{1}{\sqrt{6}} & \frac{2}{\sqrt{6}} & -\frac{1}{\sqrt{6}}\end{array}\right)\left(\begin{array}{c}1\\1\\0\end{array}\right)=\frac{3}{\sqrt{6}},$$

$$\sqrt{6}\,x=c\Rightarrow x=\frac{1}{2}.$$

The least squares solution is automatically returned by the Octave/Matlab backslash operator to solve linear systems

```
octave> A=[1; 2; -1]; b=[1; 1; 0]; x=A\b; disp(x);
   0.50000
octave>
```

(b) Through the normal equations method

$$\boldsymbol{N}=\boldsymbol{A}^T\boldsymbol{A}=\begin{pmatrix}14 & 13\\ 13 & 26\end{pmatrix}, \boldsymbol{c}=\boldsymbol{A}^T\boldsymbol{b}=\begin{pmatrix}28\\ 32\end{pmatrix}\Rightarrow$$

$$\begin{pmatrix}14 & 13 & 28\\ 13 & 26 & 32\end{pmatrix}\sim\begin{pmatrix}14 & 13 & 28\\ 0 & \frac{195}{14} & 6\end{pmatrix}\sim\begin{pmatrix}14 & 13 & 28\\ 0 & \frac{195}{14} & 6\end{pmatrix}\sim\begin{pmatrix}14 & 13 & 28\\ 0 & 1 & \frac{28}{65}\end{pmatrix}\sim\begin{pmatrix}14 & 13 & 28\\ 0 & 1 & \frac{28}{65}\end{pmatrix}\sim\begin{pmatrix}1 & 0 & \frac{8}{5}\\ 0 & 1 & \frac{28}{65}\end{pmatrix}$$

```
octave> A=[1 0; 2 -1; 3 5]; b=[1; 3; 7];
octave> N=A'*A; disp(N);
   14   13
   13   26
octave> c=A'*b; disp(c);
   28
   32
octave> x = N\c; disp([x [8./5.; 28./65.]]);
   1.60000   1.60000
   0.43077   0.43077
octave>
```

The $\boldsymbol{QR}$ factorization is found by the Gram-Schmidt procedure $(m=3, n=2)$

$$\boldsymbol{A}=(\ \boldsymbol{a}_1\ \ \boldsymbol{a}_2\ )=\begin{pmatrix}1 & 0\\ 2 & -1\\ 3 & 5\end{pmatrix}=(\ \boldsymbol{q}_1\ \ \boldsymbol{q}_2\ )\begin{pmatrix}r_{11} & r_{12}\\ 0 & r_{22}\end{pmatrix}\Rightarrow$$

$$r_{11}\boldsymbol{q}_1 = \boldsymbol{a}_1 \quad (1)$$

$$r_{12}\boldsymbol{q}_1+r_{22}\boldsymbol{q}_2 = \boldsymbol{a}_2 \quad (2)$$

Multiply (1) by $\boldsymbol{q}_1^T$, and use $\|\boldsymbol{q}_1\|^2=\boldsymbol{q}_1^T\boldsymbol{q}_1=1$, to obtain

$$r_{11}=\|\boldsymbol{a}_1\|=\sqrt{14}, \boldsymbol{q}_1=\frac{1}{r_{11}}\boldsymbol{a}_1=\frac{1}{\sqrt{14}}\begin{pmatrix}1\\ 2\\ 3\end{pmatrix}$$

Multiply (2) by $\boldsymbol{q}_1^T$, and use $\|\boldsymbol{q}_1\|^2=\boldsymbol{q}_1^T\boldsymbol{q}_1=1$, $\boldsymbol{q}_1^T\boldsymbol{q}_2=0$, to obtain

$$r_{12}=\boldsymbol{q}_1^T\boldsymbol{a}_2=\frac{1}{\sqrt{14}}(\ 1\ \ 2\ \ 3\ )\begin{pmatrix}0\\ -1\\ 5\end{pmatrix}=\frac{13}{\sqrt{14}}\Rightarrow$$

$$\boldsymbol{v}=r_{22}\boldsymbol{q}_2=\boldsymbol{a}_2-r_{12}\boldsymbol{q}_1=\begin{pmatrix}0\\ -1\\ 5\end{pmatrix}-\frac{13}{14}\begin{pmatrix}1\\ 2\\ 3\end{pmatrix}=\frac{1}{14}\begin{pmatrix}-13\\ -40\\ 31\end{pmatrix},$$

$$\|\boldsymbol{q}_2\|=1\Rightarrow r_{22}=\sqrt{\frac{195}{14}},$$

$$\boldsymbol{q}_2=\frac{1}{r_{22}}\boldsymbol{v}=\sqrt{\frac{14}{195}}\begin{pmatrix}-13\\ -40\\ 31\end{pmatrix}.$$

Solve the triangular system $\boldsymbol{R}\boldsymbol{x} = \boldsymbol{Q}^T\boldsymbol{b}$,

$$\begin{pmatrix} \sqrt{14} & \frac{13}{\sqrt{14}} \\ 0 & \sqrt{\frac{195}{14}} \end{pmatrix} \boldsymbol{x} = \begin{pmatrix} 2\sqrt{14} \\ 2\sqrt{\frac{42}{65}} \end{pmatrix} \Rightarrow \boldsymbol{x} = \begin{pmatrix} \frac{8}{5} \\ \frac{28}{65} \end{pmatrix}.$$

The above calculations can be carried out using Octave matrix/vector operations

```
octave> A=[1 0; 2 -1; 3 5]; a1=A(:,1); a2=A(:,2); b=[1; 3; 7];
octave> r11=norm(A(:,1)); disp([r11 sqrt(14.)]);
     3.7417   3.7417
octave> q1=a1/r11; r12=q1'*a2; disp([r12 13/sqrt(14.)]);
     3.4744   3.4744
octave> v=a2-r12*q1; disp(14*v');
    -13.000  -40.000   31.000
octave> r22=norm(v); disp(r22);
   3.7321
octave> q2=v/r22;
octave> R=[r11 r12; 0 r22]; Q=[q1 q2]; b=[1; 3; 7];
octave> x= R \ Q'*b; disp(x);
     1.60000
     0.43077
octave>
```

Notice that from the above, the $\boldsymbol{QR}$ procedure can lead to lengthier arithmetic operations at times, especially due to the normalization of column vectors of $\boldsymbol{Q}$. Nonetheless, this is the preferred procedure for larger matrices, and what is actually implemented in Octave through the backslash operator

```
octave> x=A\b; disp(x);
     1.60000
     0.43077
octave>
```

2. (1 course point) Textbook p.200, Exercise 4.4.4

Solution. Define the least squares problem, $\min_{\boldsymbol{c}} \|\boldsymbol{w} - \boldsymbol{A}\boldsymbol{c}\|$, $\boldsymbol{A} = \begin{pmatrix} \boldsymbol{1} & \boldsymbol{t} \end{pmatrix} \in \mathbb{R}^{m \times n}$, with $n = 2$ (linear least squares fit), $m = 8$ data points

```
octave> t=[1960 1965 1970 1975 1980 1985 1990 1995]';
octave> w=[86.0 99.8 115.8 125.0 132.6 143.1 156.3 169.5]';
octave> m=8; o=ones(m,1); A=[o t];
octave> N=A'*A; d=A'*w; c=N\d; disp(d);
     1.0281e+03
     2.0355e+06
octave>
octave>
```

The best linear fit is the line $c_1 + c_2 x$. The quality of the approximation can be assessed graphically. To this end, let us use another tool, a Gnuplot session.

```
octave> tf=(1960:1995)'; of=ones(36,1); wf=c(1)*of+c(2)*tf;
octave> plot(tf,wf,'-b',t,w,'ok');
octave> cd /home/student/courses/MATH547/homework; print -dpng hw4q2fit1.png;
octave>
```

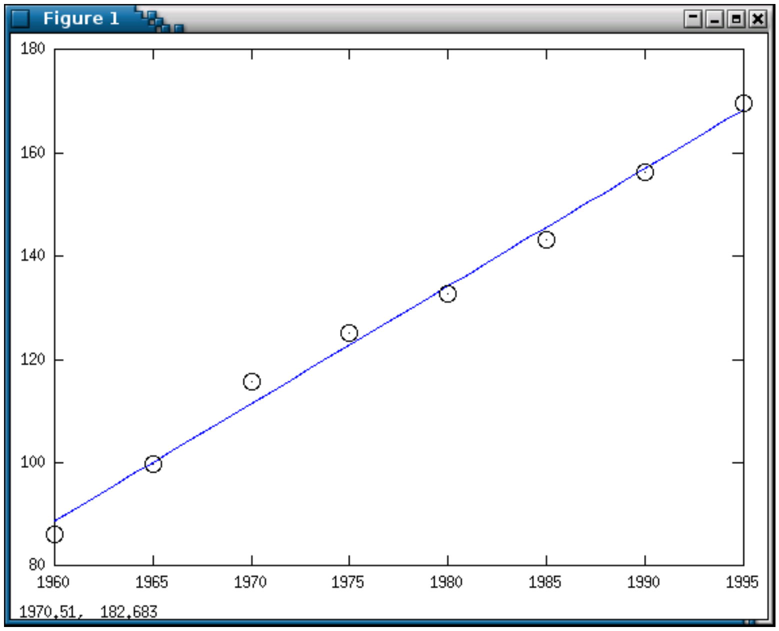

Figure 1. Linear least squares approximation of data (blue line), and original data (black circles).

The prediction for years 2000, 2005 would be

```
octave> [c(1)+c(2)*2000 c(1)+c(2)*2005]
```

(179.754   191.14)

```
octave>
```

Transform the exponential growth model $w = ce^{\alpha t}$, into a linear model by taking logarithms

$$\ln w = \ln c + \alpha t = \beta + \alpha t, \beta \equiv \ln c$$

Apply the same procedure as above for the transformed data

```
octave> lnw=log(w); A=[o t]; N=A'*A; d=A'*lnw; f=N\d; disp(f);
  -31.357138
    0.018302
octave> alpha=f(2); beta=f(1); c=exp(beta);
octave> tf=(1960:1995)'; wfe=c*exp(alpha*tf);
octave> plot(tf,wf,'-b',tf,wfe,'-r',t,w,'ok');
octave> cd /home/student/courses/MATH547/homework; print -dpng hw4q2fit2.png;
```

Graphical comparison of both fits suggests the data is more closely fitted by the linear growth model rather than the exponential growth model

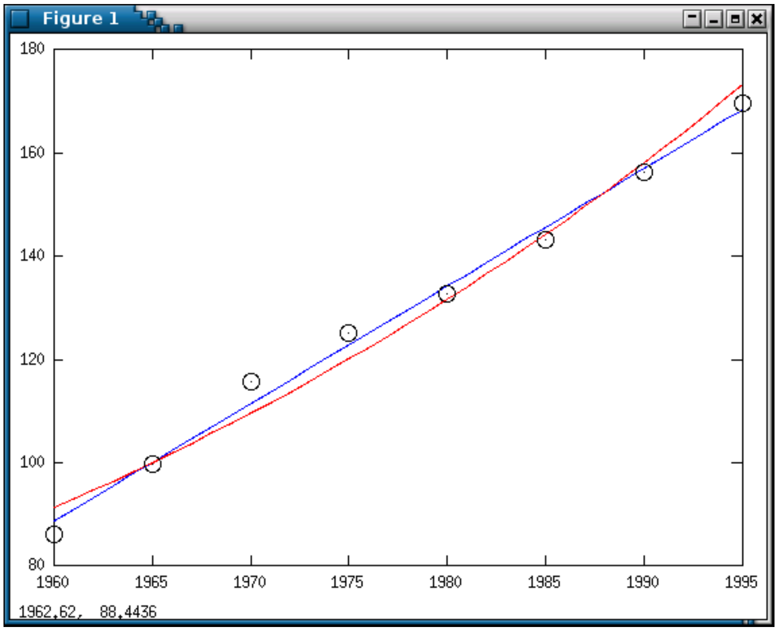

Figure 2. Linear model (blue), exponential model (red), data (black).

3. (1 course point) Textbook p.207, Exercise 4.4.13

Solution. (e) Follow procedures from Q2

```
octave> m=5; o=ones(m,1); t=[-2 -1 0 1 2]'; A=[t.^4 t.^3 t.^2 t o];
octave> y=[-1 -2 2 1 3]';
octave> c=A\y; disp(c');
    0.75000   -0.16667   -3.25000    1.66667    2.00000
octave> tf=(-2:0.2:2)'; yf=polyval(c,tf);
octave> plot(tf,yf,'b-',t,y,'ko');
octave> cd /home/student/courses/MATH547/homework; print -dpng hw4q3interpe.png;
```

Here's a comparison of the interpolation and the data

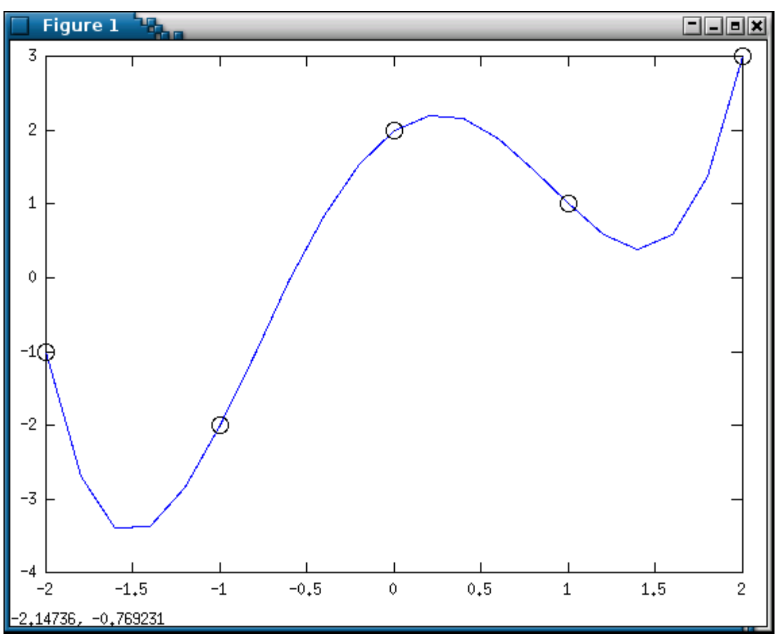

Figure 3. Interpolating polynomial (blue line), data (black circles)

4. (1 course point) Textbook p.264, Exercise 5.5.15

Solution. (c) Define the data vectors

```
octave> t=[-3. -2. -1. 0. 1. 2. 3.]'; y=[60. 80. 90. 100. 120. 120. 130.]';
octave>
```

Define the matrix of orthogonal sample vectors as defined in eq (5.71) of text

```
octave> m=max(size(t)); t0=ones(m,1); t1=t; t2=t.^2; t3=t.^3; t4=t.^4; t6=t.^6;
octave> t2av=sum(t2)/m; t4av=sum(t4)/m; t6av=sum(t6)/m;
octave> q0=t0; q1=t1; q2=t2-t2av*t0; q3=t3-(t4av/t2av)*t1;
octave> n2q0=1; n2q1=t2av; n2q2=t4av-t2av^2; n2q3=t6av-t4av^2/t2av;
octave>
```

Compute the least squares fit coefficients

```
octave> a0=q0'*y/n2q0/m; a1=q1'*y/n2q1/m; a2=q2'*y/n2q2/m; a3=q3'*y/n2q3/m;
octave> disp([a0 a1 a2 a3]);
    100.00000    11.42857    -0.95238     0.00000
octave>
```

From $a_3 = 0$ above deduce that the data might lie on a parabola. Plot the least squares fits using a finer sampling of the $t \in [-3, 3]$ interval, i.e. at step size $\Delta t = 0.2$.

```
octave> tf=(-3:0.2:3)'; mf=max(size(tf));
octave> t0f=ones(mf,1); t1f=tf; t2f=tf.^2; t3f=tf.^3; t4f=tf.^4; t6f=tf.^6;
octave> t2avf=sum(t2f)/mf; t4avf=sum(t4f)/mf; t6avf=sum(t6f)/mf;
        q0f=t0f; q1f=t1f; q2f=t2f-t2avf*t0f; q3f=t3f-(t4avf/t2avf)*t1f;
octave> yf1=a0*q0f+a1*q1f; yf2=yf1+a2*q2f;
octave> plot(tf,yf1,'-b',tf,yf2,'-g',t,y,'ok');
octave> cd /home/student/courses/MATH547/homework; print -dpng hw4q4interpc.png;
```

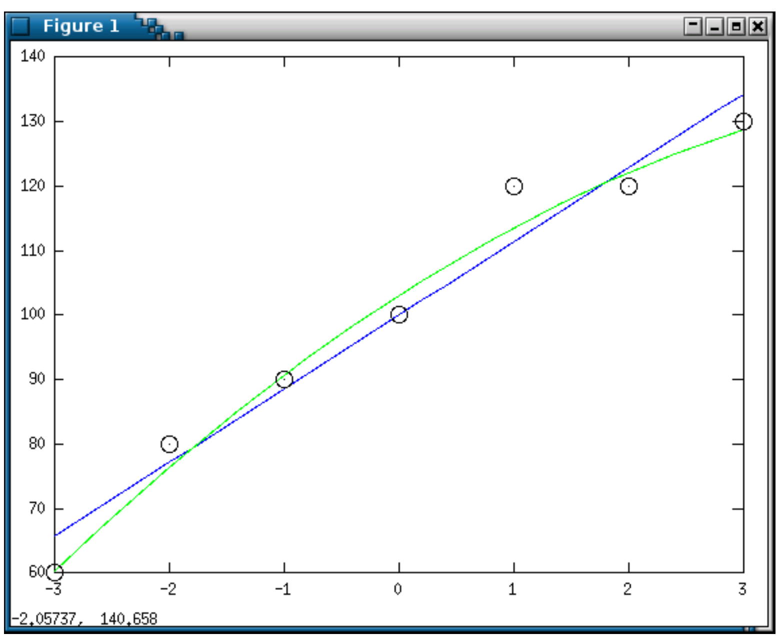

Figure 4. Least squares fit of degrees 1 (blue), 2 and 3 (green) of data (black circles). Data does not lie on a parabola.

5. (Computer application 4 course points) We consider another realistic application of linear algebra, this time using electronenchephalograms. Consult the EEG posted tutorial on how to acess data.

Task 1. (2 course points). Investigate the data for a segment that you suspect represents a large, coordinated perturbation of EEG signals, i.e., a 'wink', $\boldsymbol{w}$. Recall that a segment of length q starting at position p recorded by the i^{th} sensor is accessed as `w=data(p:p+q-1,i)`, $\boldsymbol{w} \in \mathbb{R}^q$. There is no single, "correct" choice. Make a hypothesis on choice of p, q. Let $\boldsymbol{u} \in \mathbb{R}^q$ denote some portion of the EEG data. Identify $\boldsymbol{u}$ as a possible wink if the angle between $\boldsymbol{u}$ and $\boldsymbol{w}$ is less than $\pi/90$ radians. Count the number of possible winks for each sensor channel. Then, identify a wink as occuring if a possible wink was observed in at least 3 channels in a time window of length q.

Solution. From graphical representation of data in the tutorial a possible wink might the data betwenn indices 240 and 260 in chanels 21 to 30.

```
octave> load /home/student/courses/MATH547/lessons/eeg/eeg;
octave> data=EEG.data'; [m n]=size(data); disp([m n]);

     30504       32

octave> pdata=data./max(data)+meshgrid(0:n-1,0:m-1);
octave> clf();
        hold on;
        for j=21:30
          plot(pdata(240:260,j));
        end;
        hold off;
octave> cd /home/student/courses/MATH547/homework; print -mono -deps eegwink.eps;
octave>
```

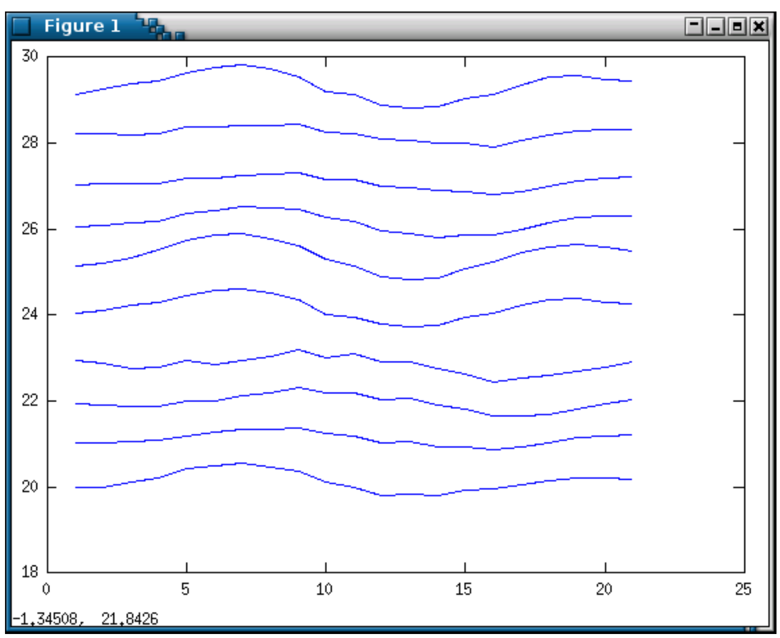

Figure 5. Possible wink chosen from channel 21, indices 240 to 255

Compute a table of angles θ between $\boldsymbol{w}$ and recordings in other channels, testing whether the other channels also record a wink to within tolerance $|\theta| < \pi/90$.

```
octave> tol=pi/90.; w=data(240:255,21);
        disp(["   Channel " "    theta " "     possible wink"]);
        for c=22:30
          u=data(240:255,c);
          theta = acos(w'*u/norm(w)/norm(u));
          if abs(theta)<tol
            ans=1;
          else
            ans=0;
          end;
          disp([c theta ans]);
        end;
      Channel     theta      possible wink
      22.00000    0.52003    0.00000
      23.00000    1.17594    0.00000
      24.00000    1.57412    0.00000
      25.00000    0.21355    0.00000
      26.00000    0.32434    0.00000
      27.00000    0.32174    0.00000
      28.00000    0.53019    0.00000
      29.00000    0.59374    0.00000
      30.00000    0.30491    0.00000
```

At tolerance $\pi/90$, no winks were observed. Try with increased tolerance $\pi/10$

```
octave> tol=pi/10.; w=data(240:255,21);
        disp(["   Channel " "    theta " "     possible wink"]);
        for c=22:30
          u=data(240:255,c);
          theta = acos(w'*u/norm(w)/norm(u));
          if abs(theta)<tol
            ans=1;
          else
            ans=0;
          end;
          disp([c theta ans]);
        end;
      Channel     theta      possible wink
      22.00000    0.52003    0.00000
      23.00000    1.17594    0.00000
      24.00000    1.57412    0.00000
      25.00000    0.21355    1.00000
      26.00000    0.32434    0.00000
      27.00000    0.32174    0.00000
      28.00000    0.53019    0.00000
      29.00000    0.59374    0.00000
      30.00000    0.30491    1.00000
octave>
```

Task 2. (2 course points). Consider n portions of brain activity at sensor position i of length q, starting from position p, i.e., `data(p:p+q-1,i)`, `data(p+q:p+2*q-1,i)`,..,`data(p+(n-1)*q, p+n*q-1)`. Organize this data as a matrix $\boldsymbol{A} \in \mathbb{R}^{q\times n}$. Ask the question: is the next sensor recording $\boldsymbol{b}$=`data(p+n*q,p+(n+1)*q-1)`, of length q indicative of normal or abnormal brain activity? You would approach this problem by investigating what part of $\boldsymbol{b} \in C(\boldsymbol{A})$, and what part $\boldsymbol{b} \in N(\boldsymbol{A}^T)$. Make various choices for p, q. Comment on your results.

Solution. Form a matrix with all n signals

```
octave> p=1; q=256; A=data(p:p+q-1,1:n);
octave>
```

Construct a loop that nulls column j of the matrix to form $\boldsymbol{A}_j$ (matrix of all signals except j), computes the projectors, and ratio of projection of column j onto $C(\boldsymbol{A}_j)$, $N(\boldsymbol{A}_j)$

```
octave> ratios=zeros(n,1);
        for j=1:n
          Aj=A; Aj(:,j)=0.;
          CAj=orth(Aj); PCAj=CAj*CAj'; PNAjT=eye(q)-PCAj;
          ratios(j) = norm(PNAjT*A(:,j))/norm(PCAj*A(:,j));
        end;
        disp([(1:n)' ratios]);
       1.000000    0.095183
       2.000000    0.260720
       3.000000    0.055983
       4.000000    0.072250
       5.000000    0.101556
       6.000000    0.235186
       7.000000    0.061580
       8.000000    0.052207
       9.000000    0.083953
      10.000000    0.136471
      11.000000    0.141196
      12.000000    0.058141
      13.000000    0.090653
      14.000000    0.074954
      15.000000    0.175634
      16.000000    0.052504
      17.000000    0.055279
      18.000000    0.059000
      19.000000    0.102820
      20.000000    0.085275
      21.000000    0.046131
      22.000000    0.064030
      23.000000    0.050734
      24.000000    0.080333
      25.000000    0.075925
      26.000000    0.069307
      27.000000    0.070883
      28.000000    0.058115
      29.000000    0.074815
      30.000000    0.043406
      31.000000    0.046306
      32.000000    0.060551
octave>
```

Using a 15% threshold, signals from sensors 2,6,15 seem different from those from the remaining sensors.