Homework 3

Due date: October 7, 2015, 11:55PM. Since multiple submissions are allowed in Sakai, submit after completing some part of the homework to avoid last minute time crunch, and/or computer failure problems.

Bibliography: Course lecture notes Lessons 13-16.

Question 1 contains an extra credit (EC3) question. You may submit the answer to EC3 with this homework or separately up to Fall Break (10/14, 11:55PM). Note: Homework 0 constitutes EC1 and EC2.

Save this file as Homework3Solution.tm in your MATH547/homework directory before starting to work on the solution.

1. (1 course points, 0.125 for each case) Use Gaussian elimination to compute the $\boldsymbol{LU}$ factorization of the following matrices. In each case explicitly show the multiplier matrix and partial result. Verify your results. You may use either Octave or hand computation to evaluate the matrix multiplications.

$$(a)\begin{pmatrix} 1 & 3 \\ -1 & 0 \end{pmatrix}; (b)\begin{pmatrix} 1 & 3 \\ 3 & 1 \end{pmatrix}; (c)\begin{pmatrix} -1 & 1 & -1 \\ 1 & 1 & 1 \\ -1 & 1 & 2 \end{pmatrix}; (d)\begin{pmatrix} 2 & 0 & 3 \\ 1 & 3 & 1 \\ 0 & 1 & 1 \end{pmatrix}; (e)\begin{pmatrix} -1 & 0 & 0 \\ 2 & -3 & 0 \\ 1 & 3 & 2 \end{pmatrix}$$

$$(f)\begin{pmatrix} 1 & 0 & -1 \\ 2 & 3 & 2 \\ -3 & 1 & 0 \end{pmatrix}; (g)\begin{pmatrix} 1 & 0 & -1 & 0 \\ 0 & 2 & -1 & -1 \\ -1 & 3 & 0 & 2 \\ 0 & -1 & 2 & 1 \end{pmatrix}; (h)\begin{pmatrix} 1 & 1 & -2 & 3 \\ -1 & 2 & 3 & 0 \\ -2 & 1 & 1 & -2 \\ 3 & 0 & 1 & 5 \end{pmatrix}; (i)\begin{pmatrix} 2 & 1 & 3 & 1 \\ 1 & 4 & 0 & 1 \\ 3 & 0 & 2 & 2 \\ 1 & 1 & 2 & 2 \end{pmatrix}$$

Background (see Lesson 13). A nonsingular matrix $\boldsymbol{A} \in \mathbb{R}^{m\times m}$ can be factorized as $\boldsymbol{LU} = \boldsymbol{A}$, with $\boldsymbol{L} \in \mathbb{R}^{m\times m}$ a lower triangular matrix, and $\boldsymbol{U} \in \mathbb{R}^{m\times m}$, an upper triangular matrix. Gaussian elimination is the factorization algorithm and can be represented in matrix form as $\boldsymbol{L}_{m-1} \cdot \ldots \cdot \boldsymbol{L}_2\boldsymbol{L}_1\boldsymbol{A} = \boldsymbol{U}$, using the lower-triangular multiplier matrices $\boldsymbol{L}_k$. Denote $\boldsymbol{L}^{-1} = \boldsymbol{L}_{m-1} \cdot \ldots \cdot \boldsymbol{L}_2\boldsymbol{L}_1$, to obtain $\boldsymbol{A} = \boldsymbol{LU}$.

Solution (a-c). In the following $\boldsymbol{L}_{m-1} \ldots \boldsymbol{L}_1\boldsymbol{A} = \boldsymbol{U}$ is shown and then $\boldsymbol{A} = \boldsymbol{LU}$. Arithmetic in Octave for (c).

$$(a)\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}\begin{pmatrix} 1 & 3 \\ -1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 3 \\ 0 & 3 \end{pmatrix} \Rightarrow \begin{pmatrix} 1 & 3 \\ -1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ -1 & 1 \end{pmatrix}\begin{pmatrix} 1 & 3 \\ 0 & 3 \end{pmatrix}$$

$$(b)\begin{pmatrix} 1 & 0 \\ -3 & 1 \end{pmatrix}\begin{pmatrix} 1 & 3 \\ 3 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 3 \\ 0 & -8 \end{pmatrix} \Rightarrow \begin{pmatrix} 1 & 3 \\ 3 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 3 & 1 \end{pmatrix}\begin{pmatrix} 1 & 3 \\ 0 & -8 \end{pmatrix}$$

```
octave> A=[-1 1 -1; 1 1 1; -1 1 2]; L1=eye(3); L1(2:3,1)=-A(2:3,1)/A(1,1); disp(L1);
      1   0   0
      1   1   0
     -1   0   1
octave> A1=L1*A; disp(A1);
     -1   1  -1
      0   2   0
      0   0   3
octave> U=A1;
```

```
octave>
```

$$(c)\begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix}\begin{pmatrix} -1 & 1 & -1 \\ 1 & 1 & 1 \\ -1 & 1 & 2 \end{pmatrix}=\begin{pmatrix} -1 & 1 & -1 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{pmatrix}\Rightarrow\begin{pmatrix} -1 & 1 & -1 \\ 1 & 1 & 1 \\ -1 & 1 & 2 \end{pmatrix}=\begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}\begin{pmatrix} -1 & 1 & -1 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{pmatrix}$$

Check in Octave:

```
octave> L=inv(L1); disp([L; A-L*U]);
      1    0    0
     -1    1    0
      1   -0    1
      0    0    0
      0    0    0
      0    0    0
octave>
```

(EC3). The algorithm is efficient since the inverse of a multiplier matrix is easily computed:

$$\boldsymbol{L}_k=\begin{pmatrix} 1 & \dots & 0 & \dots & 0 \\ 0 & \ddots & 0 & \dots & 0 \\ 0 & \dots & 1 & \dots & 0 \\ 0 & \dots & -l_{k+1,k} & \dots & 0 \\ 0 & \dots & -l_{k+2,k} & \dots & 0 \\ \vdots & \dots & \vdots & \ddots & \vdots \\ 0 & \dots & -l_{m,k} & \dots & 1 \end{pmatrix}\Rightarrow \boldsymbol{L}_k^{-1}=\begin{pmatrix} 1 & \dots & 0 & \dots & 0 \\ 0 & \ddots & 0 & \dots & 0 \\ 0 & \dots & 1 & \dots & 0 \\ 0 & \dots & l_{k+1,k} & \dots & 0 \\ 0 & \dots & l_{k+2,k} & \dots & 0 \\ \vdots & \dots & \vdots & \ddots & \vdots \\ 0 & \dots & l_{m,k} & \dots & 1 \end{pmatrix}=\boldsymbol{I}-(\boldsymbol{L}_k-\boldsymbol{I}). \tag{1}$$

(1 extra credit point) Prove the above relations (1).

Furthermore, from $\boldsymbol{L}^{-1}=\boldsymbol{L}_{m-1}\cdot\ldots\cdot\boldsymbol{L}_2\boldsymbol{L}_1$ obtain

$$\boldsymbol{L}=(\boldsymbol{L}_{m-1}\cdot\ldots\cdot\boldsymbol{L}_2\boldsymbol{L}_1)^{-1}=\boldsymbol{L}_1^{-1}\cdot\boldsymbol{L}_2^{-1}\cdot\ldots\cdot\boldsymbol{L}_{m-1}^{-1}. \tag{2}$$

The product of the inverse multiplier matrices is also easily obtained as

$$\boldsymbol{L}=\begin{pmatrix} 1 & 0 & 0 & \dots & 0 & 0 \\ l_{2,1} & 1 & 0 & \dots & 0 & 0 \\ l_{3,1} & l_{3,2} & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ l_{m-1,1} & l_{m-1,2} & l_{m-1,3} & \dots & 1 & 0 \\ l_{m,1} & l_{m,2} & l_{m,3} & \dots & l_{m,m-1} & 1 \end{pmatrix}. \tag{3}$$

(1 extra credit point) Prove relations (2) and (3).

Solution example (showing both human and computer calculations). (i) Matrix size is $m\times m=4\times 4$.

Step 1. $\boldsymbol{L}_1\boldsymbol{A}=\boldsymbol{U}_1$

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ -1/2 & 1 & 0 & 0 \\ -3/2 & 0 & 1 & 0 \\ -1/2 & 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} 2 & 1 & 3 & 1 \\ 1 & 4 & 0 & 1 \\ 3 & 0 & 2 & 2 \\ 1 & 1 & 2 & 2 \end{pmatrix}=\begin{pmatrix} 2 & 1 & 3 & 1 \\ 0 & 7/2 & -3/2 & 1/2 \\ 0 & -3/2 & -5/2 & 1/2 \\ 0 & 1/2 & 1/2 & 3/2 \end{pmatrix}$$

```
octave> A=[2 1 3 1; 1 4 0 1; 3 0 2 2; 1 1 2 2]; format rat;
```

```
octave> L1=eye(4); L1(2,1)=-1/2; L1(3,1)=-3/2; L1(4,1)=-1/2; U1=L1*A; disp(U1);
   2     1     3     1
   0   7/2  -3/2   1/2
   0  -3/2  -5/2   1/2
   0   1/2   1/2   3/2
octave>
```

Step 2. $\boldsymbol{L}_2\boldsymbol{L}_1\boldsymbol{A}=\boldsymbol{L}_2\boldsymbol{U}_1=\boldsymbol{U}_2$

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 3/7 & 1 & 0 \\ 0 & -1/7 & 0 & 1 \end{pmatrix}\begin{pmatrix} 2 & 1 & 3 & 1 \\ 0 & 7/2 & -3/2 & 1/2 \\ 0 & -3/2 & -5/2 & 1/2 \\ 0 & 1/2 & 1/2 & 3/2 \end{pmatrix}=\begin{pmatrix} 2 & 1 & 3 & 1 \\ 0 & 7/2 & -3/2 & 1/2 \\ 0 & 0 & -22/7 & 5/7 \\ 0 & 0 & 5/7 & 10/7 \end{pmatrix}$$

```
octave> L2=eye(4); L2(3,2)=3/7; L2(4,2)=-1/7; U2=L2*U1; disp(U2);
   2     1     3     1
   0   7/2  -3/2   1/2
   0     0 -22/7   5/7
   0     0   5/7  10/7
octave>
```

Step 3. $\boldsymbol{L}_3\boldsymbol{L}_2\boldsymbol{L}_1\boldsymbol{A}=\boldsymbol{L}_3\boldsymbol{U}_2=\boldsymbol{U}$

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 5/22 & 1 \end{pmatrix}\begin{pmatrix} 2 & 1 & 3 & 1 \\ 0 & 7/2 & -3/2 & 1/2 \\ 0 & 0 & -22/7 & 5/7 \\ 0 & 0 & 5/7 & 10/7 \end{pmatrix}=\begin{pmatrix} 2 & 1 & 3 & 1 \\ 0 & 7/2 & -3/2 & 1/2 \\ 0 & 0 & -22/7 & 5/7 \\ 0 & 0 & 0 & 35/22 \end{pmatrix}$$

```
octave> L3=eye(4); L3(4,3)=5/22; U=L3*U2; disp(U);
   2     1     3     1
   0   7/2  -3/2   1/2
   0     0 -22/7   5/7
   0     0     0 35/22
octave>
```

Factorization and verification. $\boldsymbol{L}^{-1}=\boldsymbol{L}_3\boldsymbol{L}_2\boldsymbol{L}_1\Rightarrow\boldsymbol{L}=(\boldsymbol{L}_3\boldsymbol{L}_2\boldsymbol{L}_1)^{-1}=\boldsymbol{L}_1^{-1}\boldsymbol{L}_2^{-1}\boldsymbol{L}_3^{-1}$, $\boldsymbol{A}=\boldsymbol{L}\boldsymbol{U}$

$$\boldsymbol{L}_1=\begin{pmatrix} 1 & 0 & 0 & 0 \\ -1/2 & 1 & 0 & 0 \\ -3/2 & 0 & 1 & 0 \\ -1/2 & 0 & 0 & 1 \end{pmatrix},\boldsymbol{L}_2=\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 3/7 & 1 & 0 \\ 0 & -1/7 & 0 & 1 \end{pmatrix}\boldsymbol{L}_3=\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 5/22 & 1 \end{pmatrix}\Rightarrow\boldsymbol{L}=\begin{pmatrix} 1 & 0 & 0 & 0 \\ 1/2 & 1 & 0 & 0 \\ 3/2 & -3/7 & 1 & 0 \\ 1/2 & 1/7 & -5/22 & 1 \end{pmatrix}$$

$$\boldsymbol{U}=\begin{pmatrix} 2 & 1 & 3 & 1 \\ 0 & 7/2 & -3/2 & 1/2 \\ 0 & 0 & -22/7 & 5/7 \\ 0 & 0 & 0 & 35/22 \end{pmatrix}$$

```
octave> I4=eye(4); L = I4-(L1-I4)-(L2-I4)-(L3-I4); disp(L);
   1    -0    -0    -0
 1/2     1    -0    -0
 3/2  -3/7     1    -0
 1/2   1/7 -5/22     1
```

```
octave> disp(L*U);
           2           1           3           1
           1           4           0           1
           3           0           2           2
           1           1           2           2
octave> disp(A-L*U);
           0           0           0           0
           0           0           0           0
           0           0           0           0
           0           0           0           0
octave>
```

2. (1 course point, 0.125 for each case) Use the Gram-Schmidt algorithm to compute the $\boldsymbol{QR}$ factorization of the matrices from Problem 1. Show intermediate results. Verify your result. You may use either Octave or hand computation to evaluate scalar products, matrix multiplications.

Background (see Lesson 14). A nonsingular matrix $\boldsymbol{A} \in \mathbb{R}^{m\times m}$ can be factorized as $\boldsymbol{QR} = \boldsymbol{A} = (\ \boldsymbol{a}_1\ \ \ldots\ \ \boldsymbol{a}_m\ )$, with $\boldsymbol{Q} = (\ \boldsymbol{q}_1\ \ \ldots\ \ \boldsymbol{q}_m\ ) \in \mathbb{R}^{m\times m}$ an orthogonal matrix, and $\boldsymbol{R} = (r_{ij})_{1\leqslant i,j\leqslant m} \in \mathbb{R}^{m\times m}$, an upper (or right) triangular matrix. The Gram-Schmidt factorization algorithm is

Algorithm (modified Gram-Schmidt)

Given m vectors $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_m$
Initialize $\boldsymbol{q}_1 = \boldsymbol{a}_1, .., \boldsymbol{q}_m = \boldsymbol{a}_m$, $\boldsymbol{R} = \boldsymbol{I}$
for $i = 1$ to m
  $r_{ii} = (\boldsymbol{q}_i^T \boldsymbol{q}_i)^{1/2}$; $\boldsymbol{q}_i = \boldsymbol{q}_i / r_{ii}$
  for $j = i+1$ to m
    $r_{ij} = \boldsymbol{q}_i^T \boldsymbol{a}_j$; $\boldsymbol{q}_j = \boldsymbol{q}_j - r_{ij}\boldsymbol{q}_i$
  end
end
return $\boldsymbol{Q}, \boldsymbol{R}$

Solution example (showing computer calculations only). (i)

$$\boldsymbol{A} = (\ \boldsymbol{a}_1\ \ \boldsymbol{a}_2\ \ \boldsymbol{a}_3\ \ \boldsymbol{a}_4\ ) = \begin{pmatrix} 2 & 1 & 3 & 1 \\ 1 & 4 & 0 & 1 \\ 3 & 0 & 2 & 2 \\ 1 & 1 & 2 & 2 \end{pmatrix}$$

Initialization: $\boldsymbol{Q} = \boldsymbol{A}, \boldsymbol{R} = \boldsymbol{I}$ (When entering a for loop in the Octave calculations below use Shift+Enter for a new line).

```
octave> format short; A=[2 1 3 1; 1 4 0 1; 3 0 2 2; 1 1 2 2]; Q=A; R=eye(4);
octave> i=1; R(i,i)=sqrt(Q(:,i)'*Q(:,i)); Q(:,i)=Q(:,i)/R(i,i);
octave> for j=i+1:4
          R(i,j)=Q(:,i)'*A(:,j); Q(:,j)=Q(:,j)-R(i,j)*Q(:,i);
        end;
octave> disp([Q R]);
   0.51640   0.06667   1.13333  -0.46667   3.87298   1.80739   3.61478   2.84019
   0.25820   3.53333  -0.93333   0.26667   0.00000   1.00000   0.00000   0.00000
   0.77460  -1.40000  -0.80000  -0.20000   0.00000   0.00000   1.00000   0.00000
```

```
      0.25820   0.53333   1.06667   1.26667   0.00000   0.00000   0.00000   1.00000
octave> i=2; R(i,i)=sqrt(Q(:,i)'*Q(:,i)); Q(:,i)=Q(:,i)/R(i,i);
octave> for j=i+1:4
          R(i,j)=Q(:,i)'*A(:,j); Q(:,j)=Q(:,j)-R(i,j)*Q(:,i);
        end;
octave> disp([Q R]);
      0.51640   0.01737   1.14027  -0.47511   3.87298   1.80739   3.61478   2.84019
      0.25820   0.92052  -0.56561  -0.18100   0.00000   3.83840  -0.39947   0.48631
      0.77460  -0.36474  -0.94570  -0.02262   0.00000   0.00000   1.00000   0.00000
      0.25820   0.13895   1.12217   1.19910   0.00000   0.00000   0.00000   1.00000
octave> i=3; R(i,i)=sqrt(Q(:,i)'*Q(:,i)); Q(:,i)=Q(:,i)/R(i,i);
octave> for j=i+1:4
          R(i,j)=Q(:,i)'*A(:,j); Q(:,j)=Q(:,j)-R(i,j)*Q(:,i);
        end;
octave> disp([Q R]);
      0.51640   0.01737   0.58698  -0.75540   3.87298   1.80739   3.61478   2.84019
      0.25820   0.92052  -0.29116  -0.04197   0.00000   3.83840  -0.39947   0.48631
      0.77460  -0.36474  -0.48682   0.20983   0.00000   0.00000   1.94262   0.47750
      0.25820   0.13895   0.57766   0.92326   0.00000   0.00000   0.00000   1.00000
octave> i=4; R(i,i)=sqrt(Q(:,i)'*Q(:,i)); Q(:,i)=Q(:,i)/R(i,i);
octave> disp([Q R]);
      0.51640   0.01737   0.58698  -0.62329   3.87298   1.80739   3.61478   2.84019
      0.25820   0.92052  -0.29116  -0.03463   0.00000   3.83840  -0.39947   0.48631
      0.77460  -0.36474  -0.48682   0.17314   0.00000   0.00000   1.94262   0.47750
      0.25820   0.13895   0.57766   0.76180   0.00000   0.00000   0.00000   1.21195
octave>
```

Verification. Since $\boldsymbol{Q}$ should be an orthogonal matrix, $\boldsymbol{Q}^T\boldsymbol{Q}$ should be the identity matrix. Since $\boldsymbol{QR}=\boldsymbol{A}$, $\boldsymbol{QR}-\boldsymbol{A}$ should be the null matrix.

```
octave> disp([Q'*Q Q*R-A]);
      1.00000   0.00000   0.00000   0.00000  -0.00000   0.00000   0.00000   0.00000
      0.00000   1.00000   0.00000  -0.00000  -0.00000   0.00000   0.00000   0.00000
      0.00000   0.00000   1.00000  -0.00000   0.00000   0.00000   0.00000   0.00000
      0.00000  -0.00000  -0.00000   1.00000  -0.00000   0.00000   0.00000   0.00000
octave>
```

3. (1 course point) Consider textbook exercises C23-26 on page 55 that were the subject of Q1&2 in Homework 2. For each of the exercises compute the projection matrix onto the column space of $\boldsymbol{A}$ and onto the left null space, project $\boldsymbol{b}$ onto the column space and left null space, and verify the nature of the solutions to $\boldsymbol{Ax}=\boldsymbol{b}$ (no solution, unique solution, infinitely many solutions). Use computer computation.

Background (see Lesson 15). Given a matrix $\boldsymbol{A}\in\mathbb{R}^{m\times n}$ of rank $k\leqslant\min(m,n)$, a reduced $\hat{\boldsymbol{Q}}\,\hat{\boldsymbol{R}}=\boldsymbol{A}$ factorization with $\hat{\boldsymbol{Q}}\in\mathbb{R}^{m\times k}$, $\hat{\boldsymbol{R}}\in\mathbb{R}^{k\times n}$ can be obtained by k steps of the Gram-Schmidt algorithm. The matrix $\boldsymbol{P}=\hat{\boldsymbol{Q}}\,\hat{\boldsymbol{Q}}^T\in\mathbb{R}^{m\times m}$ projects a vector onto the column space of $\boldsymbol{A}$. The orthogonal projector $\boldsymbol{O}=\boldsymbol{I}-\boldsymbol{P}\in\mathbb{R}^{m\times m}$ projects a vector onto the left null space.

Examples. (C21)

```
octave> A=[1 4 3 -1; 1 -1 1 2; 4 1 6 5]; b=[5 6 9]'; k=rank(A); [m n]=size(A); format
        short;
```

```
octave> disp([m n k]); Im=eye(m);
      3   4   2
octave> [Q R]=qr(A); Qhat=Q(:,1:k);
octave> P=Qhat*Qhat'; O=Im-P;
octave> bP=P*b; bO=O*b; disp([norm(bP) norm(bO) norm(bP+bO-b)]);
      1.1144e+01   4.2212e+00   8.8818e-16
octave>
```

From above results, the projection of $\boldsymbol{b}$ onto the left null space, $\boldsymbol{b_O} = \boldsymbol{O}\boldsymbol{b}$, is non-zero, hence there is no solution.

(C22)

```
octave> A=[1 -2 1 -1; 2 -4 1 1; 1 -2 -2 3]; b=[3 2 1]'; k=rank(A); [m n]=size(A);
octave> disp([m n k]); Im=eye(m);
      3   4   3
octave> [Q R]=qr(A); Qhat=Q(:,1:k);
octave> P=Qhat*Qhat'; O=Im-P;
octave> bP=P*b; bO=O*b; disp([norm(bP) norm(bO) norm(bP+bO-b)]);
      3.7417e+00   1.8853e-15   5.0877e-16
octave>
```

From above results, the projection of $\boldsymbol{b}$ onto the left null space, $\boldsymbol{b_O} = \boldsymbol{O}\boldsymbol{b}$, is zero (to machine precision), hence there is a solution. The dimension of the null space is $n - k = 1$, so there is a one-parameter, infinite family of solutions $\boldsymbol{x} = \boldsymbol{x}_p + \lambda \boldsymbol{z}_1$.

```
octave> xp=A\b; Z=null(A); disp([xp Z]);
      0.60000   0.89443
     -1.20000   0.44721
     -2.00000   0.00000
     -2.00000   0.00000
```

C23 (0.25 course points):

C24 (0.25 course points):

C25 (0.25 course points):

C26 (0.25 course points):

4. (1 course point) Consider the problem of fitting a quadratic function $f(x) = a_2 x^2 + a_1 x + a_0$ to the data $\mathcal{D} = \{(x_i, y_i), i = 1, ..., m\}$.

a) Form the sum $S(a_0, a_1, a_2) = \sum_{i=1}^{m} (y_i - f(x_i))^2$

b) Compute the derivatives

$$\frac{\partial S}{\partial a_0}, \frac{\partial S}{\partial a_1}, \frac{\partial S}{\partial a_2}$$

c) Form a system of equations $\boldsymbol{A}\boldsymbol{a} = \boldsymbol{b}$ from

$$\frac{\partial S}{\partial a_0} = 0, \frac{\partial S}{\partial a_1} = 0, \frac{\partial S}{\partial a_2} = 0$$

d) Express the matrix $\boldsymbol{A} \in \mathbb{R}^{3\times 3}$ and vector $\boldsymbol{b} \in \mathbb{R}^3$ in terms of the data $\mathcal{D}$.

Solution.

(a) The sum is

$$S(a_0, a_1, a_2) = \sum_{i=1}^{m} (y_i - a_2 x_i^2 - a_1 x_i - a_0)^2$$

(b) The derivatives are

$$\frac{\partial S}{\partial a_0} = -2\sum_{i=1}^{m} (y_i - a_2 x_i^2 - a_1 x_i - a_0)$$

$$\frac{\partial S}{\partial a_1} = -2\sum_{i=1}^{m} (y_i - a_2 x_i^2 - a_1 x_i - a_0)x_i$$

$$\frac{\partial S}{\partial a_1} = -2\sum_{i=1}^{m} (y_i - a_2 x_i^2 - a_1 x_i - a_0)x_i^2$$

(c) Setting derivatives to zero leads to equations

$$\sum_{i=1}^{m} (y_i - a_2 x_i^2 - a_1 x_i - a_0) = 0 \Rightarrow m a_0 + \left(\sum_{i=1}^{m} x_i\right)a_1 + \left(\sum_{i=1}^{m} x_i^2\right)a_1 = \sum_{i=1}^{m} y_i$$

$$\sum_{i=1}^{m} (y_i - a_2 x_i^2 - a_1 x_i - a_0)x_i = 0 \Rightarrow \left(\sum_{i=1}^{m} x_i\right) a_0 + \left(\sum_{i=1}^{m} x_i^2\right)a_1 + \left(\sum_{i=1}^{m} x_i^3\right)a_1 = \sum_{i=1}^{m} x_i y_i$$

$$\sum_{i=1}^{m} (y_i - a_2 x_i^2 - a_1 x_i - a_0)x_i^2 = 0 \Rightarrow \left(\sum_{i=1}^{m} x_i^2\right) a_0 + \left(\sum_{i=1}^{m} x_i^3\right)a_1 + \left(\sum_{i=1}^{m} x_i^4\right)a_1 = \sum_{i=1}^{m} x_i^2 y_i$$

(d) In matrix form

$$\begin{pmatrix} m & \sum_{i=1}^{m} x_i & \sum_{i=1}^{m} x_i^2 \\ \sum_{i=1}^{m} x_i & \sum_{i=1}^{m} x_i^2 & \sum_{i=1}^{m} x_i^3 \\ \sum_{i=1}^{m} x_i^2 & \sum_{i=1}^{m} x_i^3 & \sum_{i=1}^{m} x_i^4 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{m} y_i \\ \sum_{i=1}^{m} x_i y_i \\ \sum_{i=1}^{m} x_i^2 y_i \end{pmatrix}$$

5. (4 course points) Electroencephalograms (EEGs) are recordings of the electric potential on cranium skin. Research on brain activity uses EEGs to determine specific activity patterns in the brain. For example, epileptic seizures have a distinctive EEG signature. Certain emotive (fear, happiness) or cognitive (concentration, distraction) states, also show characteristic EEG signatures. Identification of such states of the brain from EEGs leads to a number of linear algebra problems. A typical EEG recording is within the lessons directory. It can loaded into Octave, and the data can be plotted.

```
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> hold on;
        for j=1:n
          plot(pdata(:,j));
        end;
        hold off;
octave> cd /home/student/courses/MATH547/homework; print -mono -deps eeg.eps;
   warning: print.m: pstoedit binary is not available.
   Some output formats are not available.
octave>
```

There are $n = 32$ electrode recordings at $m = 30504$ moments of time, and plot produced by the above instructions is in Fig. 1 (The plot was displayed into a window, and the Screenshot utility was used to capture the image).

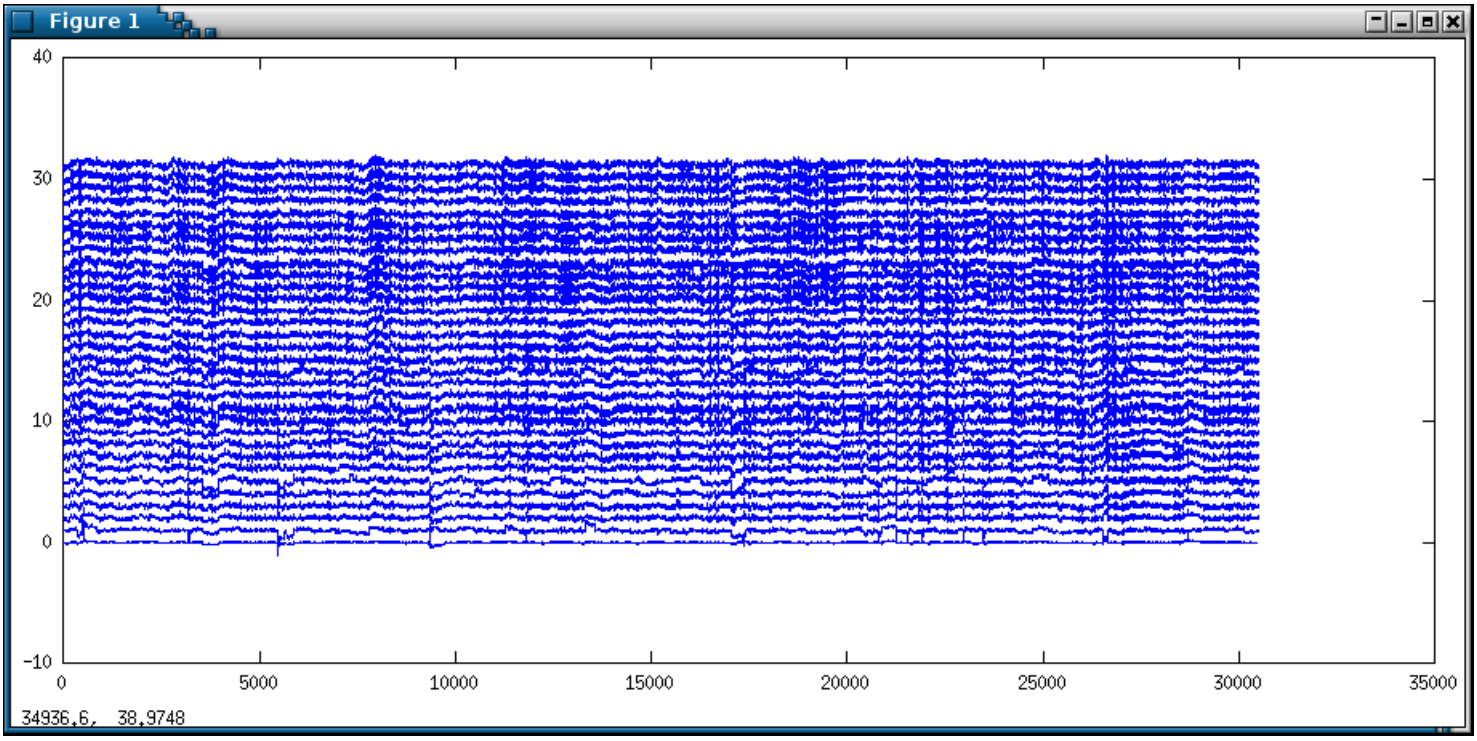

Figure 1. EEG recordings

A smaller time window of the recordings is shown in Fig. 2.

```
octave> hold on;
        for j=1:n
          plot(pdata(1001:1512,j));
        end;
        hold off;
octave> cd /home/student/courses/MATH547/homework; print -mono -deps eegwin.eps;
octave>
```

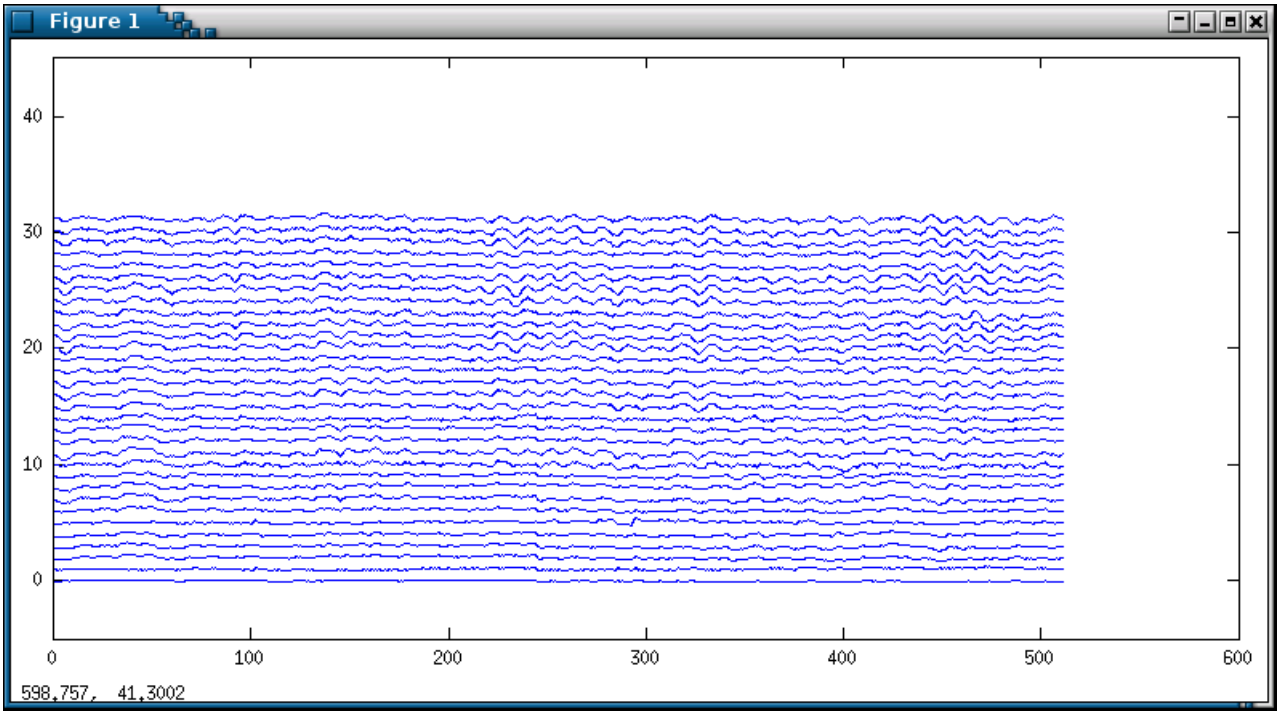

Figure 2. EEG recording from time index $i = 1001$ to $i = 1512$

5.a. There is interest if activity of the brain in a new time window is different from that in previous time windows. For electrode j, and time window width $w = 256$ we can form a matrix with p columns in which each column contains a time recording window.

```
octave> j=round(rand*32); p=100; w=128; Aj=reshape(data(1:p*w,j),w,p);
octave>
```

Construct a table of the norm of the projections of the next $q = 16$ signals (`data(p*w+1:(p+q)*w,j)`) onto the column space $C(\boldsymbol{A}_j)$ and left null space $N(\boldsymbol{A}_j^T)$. Interpret the results, asking the question: "is a subsequent signal indicative of normal activity or abnormal activity"? Repeat for two other electrodes, one close to the above ($j \pm 1$) and another one further away ($j \pm 16$ mod 32).

Solution. Form a matrix $\boldsymbol{D}$ with the data from the next 16 signals

```
octave> q=16; D=reshape(data(p*w+1:(p+q)*w,j),w,q); disp(size(D));
      128    16
octave>
```

Find the rank of $\boldsymbol{A}_j$, $r = \dim(\boldsymbol{A}_j)$

```
octave> r=rank(Aj); disp(r);
    100
octave>
```

Form bases for $C(\boldsymbol{A}_j)$, $N(\boldsymbol{A}_j^T)$

```
octave> CAj=orth(Aj); NAjT=null(Aj');
octave> disp(size(CAj));
      128   100
octave> disp(size(NAjT));
      128    28
octave>
```

Construct the projectors onto $N(\boldsymbol{A}_j^T)$ and $C(\boldsymbol{A}_j)$

```
octave> PNAjT=NAjT*NAjT'; disp(size(PNAjT)); PCAj=eye(w)-PNAjT;
      128   128
octave>
```

Construct the projections of the subsequent $q = 16$ signals onto $C(\boldsymbol{A}_j)$, $N(\boldsymbol{A}_j^T)$

```
octave> DPNAjT=PNAjT*D; DPCAj=PCAj*D;
octave>
```

Find the norms of the column vectors in the projections

```
octave> nDPNAjT=zeros(q,1); nDPCAj=zeros(q,1);
        for i=1:q
          nDPNAjT(i)=norm(DPNAjT(:,i));
          nDPCAj(i)=norm(DPCAj(:,i));
        end;
octave>
```

Display the norms, and ratio of the norm of the projection onto $C(\boldsymbol{A}_j)$ to that onto $N(\boldsymbol{A}_j^T)$.

```
octave> format short; disp([(1:q)' nDPCAj nDPNAjT nDPNAjT./nDPCAj]);
   1.0000e+00   1.9152e+02   2.2298e+01   1.1642e-01
   2.0000e+00   1.9704e+02   2.2799e+01   1.1571e-01
   3.0000e+00   1.6887e+02   2.7303e+01   1.6168e-01
   4.0000e+00   2.1041e+02   2.1341e+01   1.0142e-01
```

```
      5.0000e+00   2.4148e+02   2.5275e+01   1.0467e-01
      6.0000e+00   1.6646e+02   2.4631e+01   1.4797e-01
      7.0000e+00   1.5896e+02   3.0796e+01   1.9374e-01
      8.0000e+00   2.3411e+02   1.9480e+01   8.3207e-02
      9.0000e+00   2.2435e+02   2.1175e+01   9.4383e-02
      1.0000e+01   1.5351e+02   2.5767e+01   1.6785e-01
      1.1000e+01   2.3770e+02   1.5798e+01   6.6464e-02
      1.2000e+01   1.9534e+02   2.2591e+01   1.1565e-01
      1.3000e+01   2.3021e+02   2.4098e+01   1.0468e-01
      1.4000e+01   1.2372e+02   1.9936e+01   1.6114e-01
      1.5000e+01   1.8695e+02   2.5914e+01   1.3861e-01
      1.6000e+01   1.4147e+02   2.4706e+01   1.7463e-01
octave>
octave>
```

When the norm of the projection onto the column space $C(\boldsymbol{A}_j)$ is considerably larger than that onto $N(\boldsymbol{A}_j^T)$, the new signals are well captured by the basis set formed from the previous 100 signals. Say we set a threshold of 15%; signals 3,7,10,12,14,16 would then indicate somewhat different activity.

Repeat for data from other sensors. First a closeby sensor.

```
octave> q=16; D=reshape(data(p*w+1:(p+q)*w,j+1),w,q);
octave> DPNAjT=PNAjT*D; DPCAj=PCAj*D;
octave> nDPNAjT=zeros(q,1); nDPCAj=zeros(q,1);
        for i=1:q
          nDPNAjT(i)=norm(DPNAjT(:,i));
          nDPCAj(i)=norm(DPCAj(:,i));
        end;
octave> disp([(1:q)' nDPCAj nDPNAjT nDPNAjT./nDPCAj]);
      1.0000e+00   2.9053e+02   2.7909e+01   9.6063e-02
      2.0000e+00   2.7584e+02   2.9881e+01   1.0833e-01
      3.0000e+00   2.0552e+02   3.4558e+01   1.6815e-01
      4.0000e+00   2.6495e+02   2.2829e+01   8.6161e-02
      5.0000e+00   4.4954e+02   2.8119e+01   6.2551e-02
      6.0000e+00   3.3313e+02   2.4581e+01   7.3790e-02
      7.0000e+00   1.9171e+02   3.3039e+01   1.7234e-01
      8.0000e+00   2.7093e+02   1.7874e+01   6.5975e-02
      9.0000e+00   3.5808e+02   2.4696e+01   6.8968e-02
      1.0000e+01   2.1711e+02   2.8913e+01   1.3318e-01
      1.1000e+01   2.5038e+02   2.1627e+01   8.6376e-02
      1.2000e+01   2.8726e+02   2.6642e+01   9.2746e-02
      1.3000e+01   3.0006e+02   2.6866e+01   8.9534e-02
      1.4000e+01   2.4697e+02   2.1046e+01   8.5216e-02
      1.5000e+01   2.0243e+02   2.8139e+01   1.3901e-01
      1.6000e+01   1.9997e+02   2.8312e+01   1.4158e-01
octave>
```

With the same 15% threshold, signals 3,7 show different activity. One possible interpretation for the fewer number of signals, is that a wave is traveling from the area of the brain where sensor j is to that where sensor $j+1$ is, so that later in time activity at sensor $j+1$ is indicative of earlier in time activity at sensor j.

Repeat for a further away sensor (simple copy and paste, and modify value of j)

```
octave> q=16; D=reshape(data(p*w+1:(p+q)*w,j+16),w,q);
```

```
octave> DPNAjT=PNAjT*D; DPCAj=PCAj*D;
octave> nDPNAjT=zeros(q,1); nDPCAj=zeros(q,1);
        for i=1:q
          nDPNAjT(i)=norm(DPNAjT(:,i));
          nDPCAj(i)=norm(DPCAj(:,i));
        end;
octave> disp([(1:q)' nDPCAj nDPNAjT nDPNAjT./nDPCAj]);
      1.0000e+00   2.2552e+02   2.6774e+01   1.1872e-01
      2.0000e+00   1.6675e+02   2.2724e+01   1.3628e-01
      3.0000e+00   2.2571e+02   3.2969e+01   1.4607e-01
      4.0000e+00   1.8522e+02   2.1210e+01   1.1451e-01
      5.0000e+00   2.5051e+02   1.8557e+01   7.4077e-02
      6.0000e+00   3.0562e+02   1.7448e+01   5.7092e-02
      7.0000e+00   2.0648e+02   2.7427e+01   1.3283e-01
      8.0000e+00   1.1925e+02   1.7149e+01   1.4380e-01
      9.0000e+00   2.1974e+02   2.1018e+01   9.5650e-02
      1.0000e+01   3.1378e+02   1.7792e+01   5.6702e-02
      1.1000e+01   3.1969e+02   1.8096e+01   5.6604e-02
      1.2000e+01   3.3705e+02   1.7186e+01   5.0989e-02
      1.3000e+01   2.8145e+02   2.5487e+01   9.0556e-02
      1.4000e+01   2.6017e+02   1.4456e+01   5.5561e-02
      1.5000e+01   2.2096e+02   2.3441e+01   1.0609e-01
      1.6000e+01   2.8529e+02   1.9764e+01   6.9278e-02
octave>
```

All signals are now close to those of the original measurement of activity at sensor j w.r.t. to the 15% threshold. The precise meaning of this result depends on positions of the sensors, models of brain activity, etc.

5.b. There is interest if different electrodes show different activity. Over time window of width $w = 256$ form a matrix containing $n - 1$ signals and compute the projection of the remaining signals onto the column space and left null space. Do this for all 32 possible choices. Interpret the results to answer the question: "is the information in signal from electrode j already present in the signals from the other electrodes"?

Solution. Form a matrix with all n signals

```
octave> w=256; A=data(1:w,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(w)-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>
```

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