**Assignment 1 Solutions**

**Robotics 16-811, Fall 2015**

**Problem 1 (Implement the $PA = LDU$ decomposition algorithm directly yourself (in other words, do not just call a built-in Gaussian elimination algorithm in MatLab, for instance). You may assume that the matrix $A$ is square and invertible. Also: Do not worry about column interchanges, just row interchanges. Show that your implementation works properly.)**

Here is one possible solution, written in MatLab. Note that we are finding the LDU decomposition, not the LU decomposition. It is important to check for zero diagonal elements, so we do not divide by zero.

```
function [ L, D, U, P ] = ldu_decomp (A)

[m, n] = size(A);
p = 1:m;

for i=1:n
  A(i:m, i)
  [mv, mi] = max(abs(A(i:m, i)))
  mi = mi + i - 1;

  if mi ~= i
    display('pw')
    %Need to make a permutation
    %Swap the row
    rowtoswap = A(i , :);
    A(i , :) = A(mi , :);
    A(mi, :) = rowtoswap;
    %Swap the index
    previouspivot = p(i);
    p(i) = p(mi);
    p(mi) = previouspivot;
  end
end

L = eye(m,m);
D = eye(m,m);
U = A;

for i=1:n
  for j=(i+1):n
    if U(i, i) ~=0
      L(j, i) = U(j, i) ./ U(i, i);
    else
      L(j, i) = 0;
    end
    U(j, i:n) = U(j, i:n) - L(j, i) * U(i ,i:n);
  end
  D(i, i) = U(i, i);
  if D(i , i) ~=0
    U(i, i:n) = U(i, i:n) ./ D(i, i);
  else
    U(i, i:n) = 0;
```

```
  end
end

% Create P from indices
P = zeros(m, n);
for i=1:m
  P(i, p(i)) = 1;
end
```

## Problem 2 (Compute the PA = LDU decomposition and the SVD decomposition for each of the following matrices)

For LDU decomposition, the lu function in MatLab is used.

```
[L,U,P] = lu(A);
d = diag(U);
D = diag(d);
nonzeros = d ~= 0;
U(nonzeros,:) = U(nonzeros,:) ./ repmat(d(nonzeros), [1 size(U,2)]);
```

For SVD decomposition, the svd function in MatLab is used.

```
[U,S,V] = svd(A);
```

(a)

$$A_1 = \begin{pmatrix} 4 & 7 & 0 \\ 2 & 2 & -6 \\ 1 & 2 & 1 \end{pmatrix}$$

- LDU decomposition

$$L = \begin{pmatrix} 1 & 0 & 0 \\ 0.5 & 1 & 0 \\ 0.25 & -0.1667 & 1 \end{pmatrix}, \quad D = \begin{pmatrix} 4 & 0 & 0 \\ 0 & -1.5 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad U = \begin{pmatrix} 1 & 1.75 & 0 \\ 0 & 1 & 4 \\ 0 & 0 & 0 \end{pmatrix}, \quad P = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

- Alternative answer

$$L = \begin{pmatrix} 1 & 0 & 0 \\ 4 & 1 & 0 \\ 2 & 2 & 1 \end{pmatrix}, \quad D = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad U = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 1 & 4 \\ 0 & 0 & 0 \end{pmatrix}, \quad P = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

- SVD decomposition

$$U = \begin{pmatrix} -0.8478 & 0.4286 & -0.3123 \\ -0.4908 & -0.8571 & 0.1562 \\ -0.2008 & 0.2857 & 0.9370 \end{pmatrix}, \quad \Sigma = \begin{pmatrix} 9.0554 & 0 & 0 \\ 0 & 5.7446 & 0 \\ 0 & 0 & 0.0000 \end{pmatrix},$$

$$V = \begin{pmatrix} -0.5051 & 0.0497 & 0.8616 \\ -0.8081 & 0.3233 & -0.4924 \\ 0.3030 & 0.9450 & 0.1231 \end{pmatrix}$$

(b)

$$A_2 = \begin{pmatrix} 4 & 8 & 0 & 0 \\ 2 & 0 & -2 & 0 \\ 0 & 4 & -1 & 0 \\ 0 & -2 & 0 & 2 \\ 0 & 0 & 2 & -1 \end{pmatrix}$$

- LDU decomposition

$$L = \begin{pmatrix} 1.0000 & 0 & 0 & 0 \\ 0.5000 & 1.0000 & 0 & 0 \\ 0 & -1.0000 & 1.0000 & 0 \\ 0 & 0.5000 & -0.3333 & 1.0000 \\ 0 & 0 & -0.6667 & -0.5000 \end{pmatrix}, \quad D = \begin{pmatrix} 4 & 0 & 0 & 0 \\ 0 & -4 & 0 & 0 \\ 0 & 0 & -3 & 0 \\ 0 & 0 & 0 & 2 \end{pmatrix},$$

$$U = \begin{pmatrix} 1.0000 & 2.0000 & 0 & 0 \\ 0 & 1.0000 & 0.5000 & 0 \\ 0 & 0 & 1.0000 & 0 \\ 0 & 0 & 0 & 1.0000 \end{pmatrix}, \quad P = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

- SVD decomposition

$$U = \begin{pmatrix} -0.9005 & -0.0155 & 0.3562 & -0.1617 & -0.1895 \\ -0.0891 & -0.7311 & 0.1847 & 0.5290 & 0.3789 \\ -0.3787 & 0.0340 & -0.7352 & -0.1479 & 0.5413 \\ 0.1942 & -0.3642 & 0.2836 & -0.8023 & 0.3248 \\ 0.0078 & 0.5757 & 0.4670 & 0.1686 & 0.6496 \end{pmatrix}, \quad \Sigma = \begin{pmatrix} 9.8844 & 0 & 0 & 0 \\ 0 & 3.3506 & 0 & 0 \\ 0 & 0 & 2.3134 & 0 \\ 0 & 0 & 0 & 1.9288 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

$$V = \begin{pmatrix} -0.3824 & -0.4549 & 0.7755 & 0.2132 \\ -0.9214 & 0.2210 & -0.2847 & -0.1456 \\ 0.0579 & 0.7699 & 0.5619 & -0.2970 \\ 0.0385 & -0.3892 & 0.0434 & -0.9193 \end{pmatrix}$$

(c)

$$A_3 = \begin{pmatrix} 2 & 2 & 5 \\ 3 & 2 & 5 \\ 1 & 1 & 5 \end{pmatrix}$$

- LDU decomposition

$$L = \begin{pmatrix} 1.0000 & 0 & 0 \\ 0.6667 & 1.0000 & 0 \\ 0.3333 & 0.5000 & 1.0000 \end{pmatrix}, \quad D = \begin{pmatrix} 3.0000 & 0 & 0 \\ 0 & 0.6667 & 0 \\ 0 & 0 & 2.5000 \end{pmatrix},$$

$$U = \begin{pmatrix} 1.0000 & 0.6667 & 1.6667 \\ 0 & 1.0000 & 2.5000 \\ 0 & 0 & 1.0000 \end{pmatrix}, \quad P = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

- Alternative answer 1

$$L = \begin{pmatrix} 1 & 0 & 0 \\ 1.5 & 1 & 0 \\ 0.5 & 0 & 1 \end{pmatrix}, \quad D = \begin{pmatrix} 2 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 2.5 \end{pmatrix}, \quad U = \begin{pmatrix} 1 & 1 & 2.5 \\ 0 & 1 & 2.5 \\ 0 & 0 & 1 \end{pmatrix}, \quad P = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

- Alternative answer 2

$$L = \begin{pmatrix} 1 & 0 & 0 \\ 3 & 1 & 0 \\ 2 & 0 & 1 \end{pmatrix}, \quad D = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -5 \end{pmatrix}, \quad U = \begin{pmatrix} 1 & 1 & 5 \\ 0 & 1 & 10 \\ 0 & 0 & 1 \end{pmatrix}, \quad P = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

- SVD decomposition

$$U = \begin{pmatrix} -0.5859 & -0.0444 & -0.8091 \\ -0.6231 & -0.6138 & 0.4849 \\ -0.5182 & 0.7882 & 0.3319 \end{pmatrix}, \quad \Sigma = \begin{pmatrix} 9.7910 & 0 & 0 \\ 0 & 1.4162 & 0 \\ 0 & 0 & 0.3606 \end{pmatrix},$$

$$V = \begin{pmatrix} -0.3635 & -0.8063 & 0.4666 \\ -0.2999 & -0.3729 & -0.8781 \\ -0.8820 & 0.4591 & 0.1062 \end{pmatrix}$$

**Problem 3 ( Solve the system of equations Ax = b for the given values of A and b. Specify whether each system has zero, one, or more solutions. If the system has zero solutions give the SVD solution. If the system has a unique solution compute that solution. If the system has more than one solution specify both the SVD solution and all solutions. Relate your answers to the SVD decomposition.)**

(a)

$$A = \begin{pmatrix} 2 & 1 & 3 \\ 2 & 1 & 2 \\ 5 & 5 & 5 \end{pmatrix} \quad b = \begin{pmatrix} 5 \\ -5 \\ 0 \end{pmatrix}$$

$rank(A) = 3$ (Use the rank function in MatLab.). Thus, a unique solution exists. SVD decomposition of $A$ gives

$$U = \begin{pmatrix} -0.3635 & 0.8063 & 0.4666 \\ -0.2999 & 0.3729 & -0.8781 \\ -0.8820 & -0.4591 & 0.1062 \end{pmatrix}, \quad \Sigma = \begin{pmatrix} 9.7910 & 0 & 0 \\ 0 & 1.4162 & 0 \\ 0 & 0 & 0.3606 \end{pmatrix},$$

$$V = \begin{pmatrix} -0.5859 & 0.0444 & -0.8091 \\ -0.5182 & -0.7882 & 0.3319 \\ -0.6231 & 0.6138 & 0.4849 \end{pmatrix}$$

Hence, the unique solution for this system is given by

$$x = V\frac{1}{\Sigma}U^T b = \begin{pmatrix} -15 \\ 5 \\ 10 \end{pmatrix}$$

(b)

$$A = \begin{pmatrix} 4 & 7 & 0 \\ 2 & 2 & -6 \\ 1 & 2 & 1 \end{pmatrix} \quad b = \begin{pmatrix} 3 \\ 5 \\ 1 \end{pmatrix}$$

$rank(A) = 2$. Thus, the system either has an infinite number of solutions or no exact solution. Let us check if $b$ is in the column space of $A$. $rank([A, b]) = 3$. Since the concatenated matrix has full rank, we know that $b$ is not in the column space of $A$, and hence the system has no exact solution. SVD decomposition of $A$ gives

$$U = \begin{pmatrix} -0.8478 & 0.4286 & -0.3123 \\ -0.4908 & -0.8571 & 0.1562 \\ -0.2008 & 0.2857 & 0.9370 \end{pmatrix}, \quad \Sigma = \begin{pmatrix} 9.0554 & 0 & 0 \\ 0 & 5.7446 & 0 \\ 0 & 0 & 0.0000 \end{pmatrix},$$

$$V = \begin{pmatrix} -0.5051 & 0.0497 & 0.8616 \\ -0.8081 & 0.3233 & -0.4924 \\ 0.3030 & 0.9450 & 0.1231 \end{pmatrix}$$

Hence, the SVD solution for this system is given by

$$x = V\frac{1}{\Sigma}U^T b = \begin{pmatrix} 0.2664 \\ 0.3112 \\ -0.6205 \end{pmatrix}$$

(c)

$$A = \begin{pmatrix} 4 & 7 & 0 \\ 2 & 2 & -6 \\ 1 & 2 & 1 \end{pmatrix} \quad b = \begin{pmatrix} 18 \\ -12 \\ 8 \end{pmatrix}$$

$rank(A) = 2$. Thus, the system either has an infinite number of solutions or no exact solution. Let us check if $b$ is in the column space of $A$. $rank([A, b]) = 2$. Since the concatenated matrix does not have full rank, we know that $b$ is in the column space of $A$, and hence the system has an infinite number of solutions. SVD decomposition of $A$ gives

$$U = \begin{pmatrix} -0.8478 & 0.4286 & -0.3123 \\ -0.4908 & -0.8571 & 0.1562 \\ -0.2008 & 0.2857 & 0.9370 \end{pmatrix}, \quad \Sigma = \begin{pmatrix} 9.0554 & 0 & 0 \\ 0 & 5.7446 & 0 \\ 0 & 0 & 0.0000 \end{pmatrix},$$

$$V = \begin{pmatrix} -0.5051 & 0.0497 & 0.8616 \\ -0.8081 & 0.3233 & -0.4924 \\ 0.3030 & 0.9450 & 0.1231 \end{pmatrix}$$

Hence, the SVD solution for this system is given by

$$x_p = V\frac{1}{\Sigma}U^T b = \begin{pmatrix} 0.7879 \\ 2.1212 \\ 2.9697 \end{pmatrix}$$

The null space of $A =$

$$\text{the last } n - rank(A) \text{ columns of } V = \begin{pmatrix} 0.8616 \\ -0.4924 \\ 0.1231 \end{pmatrix}$$

Therefore, the general solution

$$x = x_p + \lambda \times \text{the null space of } A = \begin{pmatrix} 0.7879 \\ 2.1212 \\ 2.9697 \end{pmatrix} + \lambda \begin{pmatrix} 0.8616 \\ -0.4924 \\ 0.1231 \end{pmatrix}$$

where $\lambda \in \Re$.

**Problem 4 (Suppose that $u$ is an n-dimensional column vector of unit length in $\Re^n$, and let $u^T$ be its transpose. Then $uu^T$ is a matrix. Consider the $n \times n$ matrix $A = I - uu^T$ .)**

(a) **Describe the action of the matrix $A$ geometrically.**

First, note that:

$$Au = (I - uu^T)u = u - u(u^T u) = u - u \cdot 1 = 0$$
$$Au_\perp = (I - uu^T)u_\perp = u - u(u^T u_\perp) = u_\perp - u \cdot 0 = u_\perp$$

We see that $A$ projects any vector in $\Re^n$ to the ($n$ - 1) dimensional subspace orthogonal to $u$.

(b) **Give the eigenvalues of A.**

The eigenvalues will be $n-1$ 1's and a single 0.

For the 1's, we have $det(A-1I)=0$.

$$\begin{aligned} A-I &= I-uu^T-I=uu^T \\ det(uu^T) &= 0 \end{aligned}$$

because $rank(uu^T)=1$. Since the dimension of the null space of $uu^T$ is $n-1$, we have $n-1$ eigenvalues for $\lambda=1$. Geometrically, this makes sense, since there are $n-1$ basis vectors that are orthogonal to $u$, and their magnitudes should remain unchanged.

For 0 we have $det(A)=0$.

$$det(A)=det(I-uu^T)=0$$

The dimension of this null space is 1 and therefore we have 1 eigenvalue for $\lambda=0$.

(c) **Describe the null space of A.**

As shown in part (a) and (b), $Au=0$, which implies that the null space of $A$ is spanned by $u$. Geometrically, scalar multiples of $u$ projected onto the hyperplane perpendicular to $u$ gives zero.

(d) **What is $A^2$?**

$A^2=A$. Geometrically, this makes sense, because doing a projection twice is the same as doing it once.

$$A^2=(I-uu^T)(I-uu^T)=II-2Iuu^T+u(u^Tu)u^T=I-2uu^T+u(1)u^T=I-uu^T$$

**Problem 5 (The following problem arises in a large number of robotics and vision problems: Suppose $p_1,...,p_n$ are the 3D coordinates of n points located on a rigid body in three-space. Suppose further that $q_1,...,q_n$ are the 3D coordinates of these same points after the body has been translated and rotated by some unknown amount. Devise and demonstrate an algorithm in which SVD plays a central role for inferring the body's translation and rotation. Comment: your algorithm should make use of all the information available. True, you only need three pairs of points but if you use more points your solution will be more robust, something that might come in handy some day when you need to do this for real with noisy data.**
**Hint: Think inertia and/or correlation matrices.)**

Let

$$P=\begin{pmatrix} | & | & \cdots & | \\ p_1 & p_2 & \cdots & p_n \\ | & | & \cdots & | \end{pmatrix}$$

be the $n$ initial 3D coordinates, and let

$$Q=\begin{pmatrix} | & | & \cdots & | \\ q_1 & q_2 & \cdots & q_n \\ | & | & \cdots & | \end{pmatrix}$$

be the corresponding final coordinates (after rotation and translation).

We'd like to find the rotation matrix $R$ and translation vector $t$ such that $q_i = Rp_i + t \quad \forall i$. However, since it's likely there is noise in our points, we will instead find the $R$ and $t$ that minimize:

$$C = \sum_{i=1}^{n} ||Rp_i + t - q_i||^2$$

Let's differentiate this expression with respect to $t$ and set it equal to 0 to find the minimum:

$$\begin{aligned}
0 = \frac{\partial C}{\partial t} &= \frac{\partial}{\partial t} \sum_{i=1}^{n} (Rp_i + t - q_i)^T (Rp_i + t - q_i) \\
0 &= 2 \sum_{i=1}^{n} (Rp_i + t - q_i) \\
0 &= \sum_{i=1}^{n} (Rp_i - q_i) + nt \\
t &= \frac{1}{n} \sum_{i=1}^{n} (q_i - Rp_i) \\
t &= \frac{1}{n} \sum_{i=1}^{n} q_i - R \frac{1}{n} \sum_{i=1}^{n} p_i \\
t &= \bar{q} - R\bar{p}
\end{aligned}$$

Since $\frac{\partial^2 C}{\partial t^2} = n > 0$, this expression minimizes $C$. So we see that for a given R, the $t$ that minimizes our overall cost function is simply the translation between centroids, where $p$ is rotated into $q$'s frame first. Intuitively, this means that we expect the centroids of our 2 point clouds to correspond to each other. Plugging this expression back into our original cost function, we get:

$$\begin{aligned}
C &= \sum_{i=1}^{n} ||Rp_i + t - q_i||^2 \\
&= \sum_{i=1}^{n} ||Rp_i + \bar{q} - R\bar{p} - q_i||^2 \\
&= \sum_{i=1}^{n} ||R(p_i - \bar{p}) - (q_i - \bar{q})||^2 \\
&= \sum_{i=1}^{n} ||Rp_i' - q_i'||^2
\end{aligned}$$

Where $p_i' = p_i - \bar{p}$ and $q_i' = q_i - \bar{q}$ give the relative initial and final positions for each of the $n$ points. At this point, let's define:

$$P' = \begin{pmatrix} | & | & \cdots & | \\ p_1' & p_2' & \cdots & p_n' \\ | & | & \cdots & | \end{pmatrix} \quad \text{and} \quad Q' = \begin{pmatrix} | & | & \cdots & | \\ q_1' & q_2' & \cdots & q_n' \\ | & | & \cdots & | \end{pmatrix}$$

Once we have transformed our points in this way, we expect to be able to find correspondences with only a rotation. Let's now find the rotation matrix $R$ that minimizes $\sum_{i=1}^{n} ||Rp'_i - q'_i||^2$:

$$\begin{aligned}
\arg\min_R \sum_{i=1}^{n} ||Rp'_i - q'_i||^2 &= \arg\min_R \sum_{i=1}^{n} (Rp'_i - q'_i)^T (Rp'_i - q'_i) \\
&= \arg\min_R \sum_{i=1}^{n} ((Rp'_i)^T Rp'_i - (Rp'_i)^T q'_i - q_i'^T Rp'_i - q_i'^T q'_i) \\
&= \arg\min_R \sum_{i=1}^{n} (p_i'^T R^T Rp'_i - Rp_i'^T R^T q'_i - q_i'^T Rp'_i - q_i'^T q'_i) \\
&= \arg\min_R \sum_{i=1}^{n} (-p_i'^T R^T q'_i - q_i'^T Rp'_i) \\
&= \arg\max_R \sum_{i=1}^{n} (p_i'^T R^T q'_i + p_i'^T R^T q'_i) \\
&= \arg\max_R \sum_{i=1}^{n} p_i'^T R^T q'_i \\
&= \arg\max_R \sum_{i=1}^{n} Tr(Rp'_i q_i'^T) \\
&= \arg\max_R Tr(RP'Q'^T)
\end{aligned}$$

To find the best $R$, we need to maximize this trace. If we take the SVD of $P'Q'^T$, we get:

$$\begin{aligned}
Tr(RP'Q'^T) &= Tr(RU\Sigma V^T) \\
&= Tr(RUV^T\Sigma) \\
&= Tr(M\Sigma) \\
&= \sum m_{ii}\sigma_{ii}
\end{aligned}$$

Since $M = RUV^T$ is orthogonal, $|m_{ii}| \leq 1$, and we also know that $\sigma_{ii} \geq 0$, so we get that the sum is maximized by $m_{ii} = 1, \forall i$. Therefore, $M = I$ and $R^* = VU^T$.

A sample MatLab program implementing this algorithm is on the next page.

```
%
% find_transform.m
%
% Finds the best rigid body transform between 2 corresponding sets of
% points in a least-squares sense. For the algorithm for how to compute the
% best transform, please refer to the solution set for 16-811 Homework 1,
% Problem 5.
%
% Input: P, a 3 x N matrix containing the initial points in 3d
%        Q, a 3 x N matrix containing the transformed points in 3d
%
% Output: R, the best 3 x 3 rotation matrix
%         t, the best 3 x 1 translation vector
%
% With no noise, Q = R * P + t. With noise, we minimize ||R*P+t - Q||^2.

function [R, t] = find_transform(P, Q)

% Find the centroids of our data
p_bar = mean(P, 2);
q_bar = mean(Q, 2);

% Transform all of the points relative to their centroid
Pp = bsxfun(@minus, P, p_bar);
Qp = bsxfun(@minus, Q, q_bar);

% Find the svd of the correlation matrix
[U, ~, V] = svd(Pp * Qp');

% Compute our rotation matrix according to our algorithm
R = V * U';

% Compute our translation matrix according to our algorithm
t = q_bar - R * p_bar;

end
```