Numerical Analysis

Assignment 2, October 2017

Kamyar Nazeri
Student ID: 100633486

Question 1.

Matrix $A \in \mathbb{C}^{n\times n}$ is said to be *strictly column diagonally dominant* if

$$|a_{j,j}| > \sum_{\substack{i=1\\ i\neq j}} |a_{i,j}| \quad (j = 1 : n) \tag{1}$$

Prove that a strictly column diagonally dominant matrix is always nonsingular.

Suppose A is singular, that is $det(A) = 0$; then $A\mathbf{x} = 0$ has non-trivial solution and there exists some non-zero vector $\mathbf{u} = (u_1, u_2, ..., u_n)^T$ such that

$$A\mathbf{u} = 0 \tag{2}$$

By taking the transpose of equation (2):

$$\mathbf{u}^T A^T = 0 \tag{3}$$

Let k be the index where

$$u_k \geq u_i \quad \text{for all} \quad i = 1, 2..., n$$

From the k-th row of equation (3), we obtain:

$$a_{(k,1)}u_1 + a_{(k,2)}u_2 + ... + a_{(k,k)}u_k + ... + a_{(k,n)}u_n = 0 \tag{4}$$

$$a_{(k,1)}u_1 + a_{(k,2)}u_2 + ... + a_{(k,n)}u_n = -a_{(k,k)}u_k$$

Hence:

$$\left|a_{(k,k)}u_k\right| = \left|\sum_{i\neq k} a_{(k,i)}u_i\right| = \sum_{i\neq k} \left|a_{(k,i)}u_i\right| \tag{5}$$

$$\left|a_{(k,k)}\right| = \sum_{i\neq k} \left|a_{(k,i)}\frac{u_i}{u_k}\right| = \sum_{i\neq k} \left|a_{(k,i)}\right| \left|\frac{u_i}{u_k}\right|$$

Since $\left|\frac{u_i}{u_k}\right| \leq 1$ for every i:

$$\left|a_{(k,k)}\right| \leq \sum_{i\neq k} \left|a_{(k,i)}\right| \tag{6}$$

Which is a contradiction with (1). Therefore assumption (2) is incorrect and $A\mathbf{x} = 0$ has only trivial solution and A is nonsingular.

Question 2.

Write a program to construct a least-squares polynomial fitting certain data by distinct computational methods. In particular, find the least-squares polynomial of degree 11 (i.e., $n = 12$) fitting $m = 50$ points obtained by sampling the function $f(x) = cos(4x)$ at m points uniformly spaced across the interval [0, 1]. That is, the programs should produce the 12 coefficients of the least-squares polynomial obtained by "solving" the over-determined linear system of equations.

(a) Write a program to compute the coefficient matrix A and the right-hand side vector $\mathbf{f}$.

```
n = 12;                  % polynomial degree: (n-1)
m = 200;                 % number of samples

x = linspace(0, 1, m);   % sample m points on the interval [0,1]
f = cos(4 * x)';         % evaluate function on sample points
A = fliplr(vander(x));   % create the Vandermonde Matrix
A = A(:,1:n);            % get polynomials up to (n-1)
```

(b) Modify your program from part (a) to compute the coefficients $\mathbf{c}^{(Chol)}$ obtained when solving the normal equations $A^T A\mathbf{c} = A^T\mathbf{f}$ by computing the Cholesky decomposition of $A^T A$

```
% to solve the overdetermined system of equations Ac = f
% we calculate Cholesty decomposition of A'A and get A'A = LL'
% multiply both sides of the equation by A' => A'Ac=A'f
% to solve for c: c_chol = inv(LL')A'f

[L] = chol(A'*A,'lower');   % lower triangular matrix L
c_chol = L' \ (L \ (A'*f));
```

(c) Modify your program from part (a) to compute the coefficients $\mathbf{c}^{(QR)}$ obtained using the QR decomposition of A.

```
% to solve the overdetermined system of equations Ac = f
% we calculate QR decomposition of A and get A = QR
% to solve for c: c_qr = inv(R)*Q'*f

[Q,R] = qr(A,0);            % reduced QR factors
c_qr = R\(Q'*f);            % inv(Q)=Q'
```

(d) Modify your program from part (a) to compute the coefficients $\mathbf{c}^{(SVD)}$ obtained using the singular value decomposition of A.

```
% to solve the overdetermined system of equations Ac = f
% we calculate SVD decomposition of A and get A = USV
% to solve for c: c_svd = inv(USV)f

[U,S,V] = svd(A,0);         % performs a svd of matrix A
c_svd = V / S * U' * f;     % inv(U)=U' and inv(V')=V
```

The code is available as a MATLAB file (*question2.m*)

(e) Plot the absolute residuals $|\mathbf{r} - A\mathbf{c}|$ for each of the three solutions $\mathbf{c}$ computed against x.

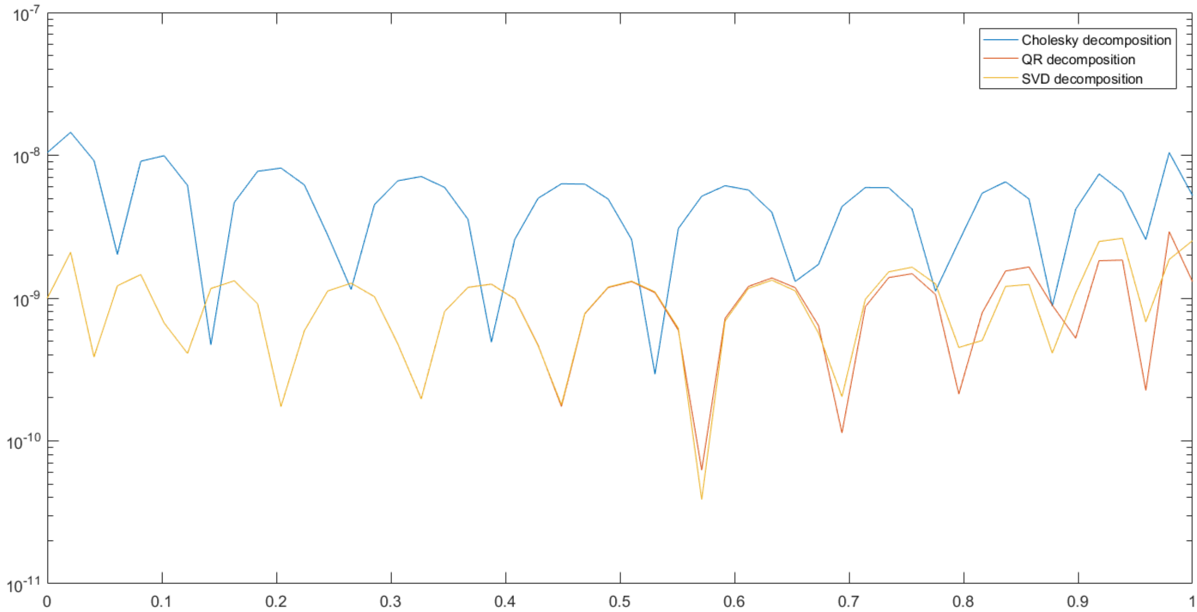

Figure 1: Residuals computed for each of the coefficient vectors $\mathbf{c}^{(Chol)}$, $\mathbf{c}^{(QR)}$, and $\mathbf{c}^{(SVD)}$.

(f) Print a table to 16 digits of precision showing the coefficients of the least-squares polynomials:

n	**Cholesky decomposition**	**QR decomposition**	**SVD decomposition**
0	0.999999989561589	1.000000000996608	1.000000000996610
1	0.000002624202733	-0.000000422743112	-0.000000422743110
2	-8.000090615741891	-7.999981235684255	-7.999981235684690
3	0.001236828943967	-0.000318763244498	-0.000318763245417
4	10.657870472602468	10.669430795958332	10.669430795960118
5	0.037032551084895	-0.013820288155987	-0.013820288089856
6	-5.788074905310304	-5.647075627080019	-5.647075626867935
7	0.177871226754187	-0.075316024565097	-0.075316024316973
8	1.399642921099546	1.693606963572196	1.693606963868704
9	0.219057608697183	0.006032108788395	0.006032109701667
10	-0.461840020342716	-0.374241703484410	-0.374241703992312
11	0.103647702805936	0.088040576080084	0.088040576075623

Coefficients of the least-squares polynomials using Cholesky, QR, & SVD decomposition.

Highlighted coefficients are those from Cholesky decomposition, that are very different from QR and SVD decompositions.

Question 3.

(a) Let $A \in \mathbb{C}^{n\times n}$ be an outer product of two vectors, i.e., $A = \mathbf{y}\mathbf{x}^H$ for some vectors $\mathbf{x}, \mathbf{y} \in \mathbb{C}^n$. Show that $\|A\|_2 = \|\mathbf{x}\|_2 \|\mathbf{y}\|_2$, i.e., the (matrix) ℓ_2-norm of A is the product of the (vector) ℓ_2-norm of $\mathbf{x}$ and $\mathbf{y}$.

$$AA^H = \mathbf{y}\mathbf{x}^H\mathbf{x}\mathbf{y}^H \tag{1}$$

$\mathbf{x}$ is a vector in $\mathbb{C}^n \quad \Rightarrow \quad \mathbf{x}^H\mathbf{x} = \|\mathbf{x}\|_2^2$

$$AA^H = \mathbf{y}\,\|\mathbf{x}\|_2^2\,\mathbf{y}^H = \|\mathbf{x}\|_2^2\,\mathbf{y}\mathbf{y}^H \tag{2}$$

We define $\rho(B) := \max\{\textit{absolute value of eigenvalues of } B\}$

$$\sqrt{\rho(AA^H)} = \sqrt{\rho(\|\mathbf{x}\|_2^2\,\mathbf{y}\mathbf{y}^H)} \tag{3}$$

We know that:

$$A\mathbf{v} = \lambda\mathbf{v} \ (\lambda \text{ eigenvalue of } A) \quad \Rightarrow \quad cA\mathbf{v} = c\lambda\mathbf{v} \ (c\lambda \text{ eigenvalue of } cA \ \forall\ c\ \neq 0 \in\ \mathbb{R})$$

Equation (3) becomes:

$$\sqrt{\rho(AA^H)} = \sqrt{\|\mathbf{x}\|_2^2\,\rho(\mathbf{y}\mathbf{y}^H)} \tag{4}$$

Because $\|\mathbf{x}\|_2 > 0$

$$\sqrt{\rho(AA^H)} = \|\mathbf{x}\|_2\,\sqrt{\rho(\mathbf{y}\mathbf{y}^H)} \tag{5}$$

Which by definition of ℓ_2 norm becomes:

$$\|A\|_2 = \|\mathbf{x}\|_2\,\|\mathbf{y}\|_2 \tag{6}$$

(b) Let $K_p(A) = \|A\|_p \left\|A^{-1}\right\|_p$ be the condition number of matrix inversion in the p-norm. If $A \in \mathbb{C}^{n\times n}$ and $B \in \mathbb{C}^{n\times n}$ are both invertible, prove that $K_p(AB) \leq K_p(A)K_p(B)$.

A and B are invertible then $AB \in \mathbb{C}^{n\times n}$ is invertible:

$$K_p(AB) = \|AB\|_p \left\|(AB)^{-1}\right\|_p = \|AB\|_p \left\|B^{-1}A^{-1}\right\|_p \tag{1}$$

A and B are square matrices and all matrix p-norms are consistent:

$$\|AB\| \leq \|A\|\,\|B\|$$

Equation (1) becomes:

$$K_p(AB) = \|AB\|_p \left\|B^{-1}A^{-1}\right\|_p \leq \|A\|\,\|B\|\,\left\|B^{-1}\right\|\,\left\|A^{-1}\right\| \tag{2}$$

$$K_p(AB) \leq \|A\|\,\left\|A^{-1}\right\|\,\|B\|\,\left\|B^{-1}\right\|$$

$$K_p(AB) \leq K_p(A)K_p(B)$$

Question 4.

The determinant of a triangular matrix is the product of its diagonal entries. Use this fact to develop a MATLAB function `ludet` for computing the determinant of any arbitrary $A \in \mathbb{C}$ using its LU decomposition.

We implement LU decomposition using MATLAB's `lu` command. The determinant is the product of the entries on the diagonal of the U matrix.
To determine the sign of the determinant, we count the number of permutations in the *permutation* matrix returned by `lu` function. The following MATLAB function shows the implementation:

```
function d = ludet(A)
  [l, u, p] = lu(A);

  sign = 1;
  [row, col] = find(p);

  % counting the number of permutations
  for i=1:size(p)
    if(row(i) ~= i)
      j = find(row == i);  % other row index
      r = row(i);          % current row
      q = p(i,:);          % current permutation row

      row(i) = i;          % update current row
      p(i,:) = p(r,:);     % update current permutation row

      row(j) = r;          % update other row
      p(row(i),:) = q;     % update other permutation row

      sign = -sign;
    end
  end

  d = sign * prod(diag(u));
```

The code is available as a MATLAB file (*ludet.m*) and a MATLAB diary (*ludet.txt*)