Chapter 2

Linear Systems

2.1 Gaussian Elimination with Partial Pivoting

The most commonly occurring problems in numerical analysis are the solution of a linear system

$$A\mathbf{x} = \mathbf{b}.$$

Here A is $m \times n$ (real) matrix and $\mathbf{b} \in \mathbb{R}^n$, but in most cases we assume A is a square matrix.

Example 2.1.1. We write the system in the form of $[A : \mathbf{b}]$ and reduce it to the form $[U : \tilde{\mathbf{b}}]$ (U is upper triangular) by **elementary row operations**:

(1) Subtracting a multiple of i-th row from j-th row.

(2) Interchange rows i and j

(3) Multiplying a row by a nonzero scalar.

These operations leave the system set unchanged and U is said to be **(row) equivalent**. $A \sim U$.

Assume $A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ and $\mathbf{b} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$.

$$\begin{bmatrix} 1 & 2 & :2 \\ 3 & 4 & :3 \end{bmatrix} \sim \begin{bmatrix} 1 & 2 & : & 2 \\ 0 & -2 & : & -3 \end{bmatrix}.$$

The elementary row operation involved here is $R_2 \leftarrow R_2 - \frac{1}{3}R_1$. The solution of $A\mathbf{x} = \mathbf{b}$ is obtained by **back substitution**.

Round off error amplified if divided by a small number

Problem with finite precision. Assume a computer can store only five decimal digits.

Example 2.1.2.

$$a = \frac{1}{6} = 0.166666666..... \text{ will be stored as } a = 0.16667$$

Now compute $\frac{10.001}{a}$. Its exact value is 60.006. However, in our computer it will be computed as

$$\frac{10.001}{0.16667} = 60.0047999 = 60.00480$$

An absolute error is 0.0012. A relative error of $0.0012/60.006 = 0.00002$. Next consider

$$b = \frac{1}{6000} = 0.00016666..... \text{ will be stored as } 0.00017$$

Now compute $\frac{10.001}{b}$, whose exact value is 60006. But the computer arithmetic gives

$$\frac{10.001}{0.00017} = 58829.41177.$$

An absolute error is 1176.5882. A relative error of $1176.5882/60006 = 0.0196078$.

Try dividing by a larger number:

Example 2.1.3. Let $a = 100/6 = 16.66666... \approx 16.66667$ Now compute $\frac{10.001}{a}$. Its exact value is 0.60006. In our computer it will be computed as

$$\frac{10.001}{16.66667} = 0.60005988$$

A relative error of

$$0.00000012/0.6 = 0.0000002$$

Why? Dividing by small number may induce relatively larger error due to round off error(Finite precision). In fact the relative error of

$$1/6000 \approx 0.00017$$

is

$$(0.00017 - 0.0001666666)/0.000166666 = 0.00000333/0.00016666 = 0.02$$

the relative error of $1/6 \approx 0.16667 = 0.00002$ is

$$(0.16667 - 0.166666666)/0.16666666 = 0.00002$$

The relative error of $100/6 \approx 16.66667$ is

$$(16.66667 - 16.6666666)/16.6666666 = 0.0000002$$

much smaller!

Examples in the book

Multiplier, Pivot, Pivot Position

Gaussian Elimination with Partial Pivoting

1. For $i = 1, \cdots, n-1$.
2. Find the largest entry in column i from row i to row n.
   If the largest value is zero, signal(flag) unique solution does not exists and stop.
3. If necessary, perform a row interchange
4. For $j = i+1$ to n perform $R_j \leftarrow R_j - m_{j,i}R_i$ where $m_{j,i} = a_{j,i}/a_{i,i}$.
5. If the (n,n) entry is zero, signal that a unique solution does not exist and stop.
   Otherwise, solve the system by back substitution

Example 2.1.4. Use G-elim to solve the system $A\mathbf{x} = \mathbf{b}$ where $A = [122; 4412; 4812]$ and $\mathbf{b} = [1, 12, 8]^T$

$$\begin{bmatrix} 1 & 2 & 2 & :1 \\ 4 & 4 & 12 & :12 \\ 4 & 8 & 12 & :8 \end{bmatrix} \sim \begin{bmatrix} 4 & 4 & 12 & :12 \\ 1 & 2 & 2 & :1 \\ 4 & 8 & 12 & :8 \end{bmatrix}$$

$$\sim \begin{bmatrix} 4 & 4 & 12 & :12 \\ 0 & 1 & -1 & :-2 \\ 0 & 4 & 0 & :-4 \end{bmatrix}$$

$$\sim \begin{bmatrix} 4 & 4 & 12 & :12 \\ 0 & 4 & 0 & :-4 \\ 0 & 1 & -1 & :-2 \end{bmatrix}$$

$$\sim \begin{bmatrix} 4 & 4 & 12 & :12 \\ 0 & 4 & 0 & :-4 \\ 0 & 0 & -1 & :-1 \end{bmatrix}$$

Multiplier: $m_{21} = 1/4$, $m_{31} = 1$, $m_{32} = 1/4$

$$\begin{bmatrix} 1 & 0 & 0 \\ m_{21} & 1 & 0 \\ m_{31} & m_{32} & 0 \end{bmatrix}$$

A **complete pivoting** is possible but it is rarely used.

2.2 LU-Decomposition

Apply the Gaussian Elimination to

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$$

Step 1- Eliminating the first column

Assume $a_{11} \neq 0$. Then by an elementary row operation E_{21}, we have

$$E_{21}A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ 0 & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$$

Here

$$E_{21} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ -\frac{a_{21}}{a_{11}} & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}$$

Similarly with

$$E_{31} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ -\frac{a_{31}}{a_{11}} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}$$

we obtain

$$E_{31}E_{21} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ -\frac{a_{21}}{a_{11}} & 1 & \cdots & 0 \\ -\frac{a_{31}}{a_{11}} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}.$$

Repeat this process

$$E_{n1} \cdots E_{31}E_{21}A = A^{(2)}.$$

Let $L_1 = E_{n1} \cdots E_{21}$. Then

$$L_1 A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ 0 & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} \\ 0 & \vdots & \vdots & \vdots \\ 0 & a_{n2}^{(2)} & \cdots & a_{nn}^{(2)} \end{bmatrix} = A^{(2)}.$$

Step 2- Eliminating the second column

We repeat similar process to the $(n-1) \times (n-1)$ submatrix of $A^{(2)}$. Hence we can find $L_2 = E_{n2} \cdots E_{32}$ such that

$$L_2 A^{(2)} = \begin{bmatrix} a_{11} & a_{12} & & \cdots & a_{1n} \\ 0 & a_{22}^{(2)} & & \cdots & a_{2n}^{(2)} \\ 0 & 0 & a_{33}^{(3)} & \cdots & a_{3n}^{(3)} \\ 0 & 0 & \vdots & \vdots & \vdots \\ 0 & 0 & a_{3n}^{(3)} & \cdots & a_{nn}^{(3)} \end{bmatrix}.$$

Step 3- Eliminating the third column and more

We repeat step 2 to eliminate all the entries below diagonal to have

$$L_{n-1} \cdots L_2 L_1 A = U$$

where

$$L = L_1^{-1} \cdots L_{n-1}^{-1}. \tag{2.1}$$

Entries of L_1 is $-m_{i1} = -\frac{a_{i1}}{a_{11}}$, $(2 \leq i \leq n)$. Similarly the entries of L_k is $-m_{i,k} = -\frac{a_{i,k}^{(k)}}{a_{k,k}^{(k)}}$, $(k+1 \leq i \leq n)$. Thus

$$L_1 = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ -m_{21} & 1 & \cdots & 0 \\ -m_{31} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ -m_{n1} & 0 & \cdots & 1 \end{bmatrix}, \quad L_1^{-1} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ m_{21} & 1 & \cdots & 0 \\ m_{31} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ m_{n1} & 0 & \cdots & 1 \end{bmatrix}$$

$$L_2 = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & -m_{32} & 1 & \cdots & 0 \\ 0 & -m_{42} & 0 & 1\cdots & 0 \\ . & \cdots & \cdots & \cdots & 0 \\ 0 & -m_{n2} & 0 & \cdots & 1 \end{bmatrix}, \quad L_1^{-1}L_2^{-1} = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ m_{21} & 1 & \cdots & 0 & 0 \\ m_{31} & m_{32} & \cdots & 0 & 0 \\ . & \cdots & \cdots & \cdots & 0 \\ m_{n1} & m_{n2} & \cdots & 0 & 1 \end{bmatrix}$$

L is unit lower triangular(sometimes called Doolittle method) and U is formed as follows: Let A_1, A_2 denote the rows of A. Then $U_1 = A_1$, $U_2 = A_2 - m_{21}A_1$. Hence the rows of U are linear combinations of rows of A. Repaeting

$$A = A^{(1)} \to A^{(2)} \to \cdots \to A^{(n-1)}.$$

This continues as long as pivot entry is nonzero.

$$a_{ij}^{(k+1)} = \begin{cases} a_{ij}^{(k)} & \text{if } i \leq k \\ a_{ij}^{(k)} - \left(a_{ik}^{(k)}/a_{kk}^{(k)}\right) a_{kj}^{(k)} & \text{if } i \geq k+1 \text{ and } j \geq k+1 \\ 0 & \text{if } i \geq k+1 \text{ and } j \leq k. \end{cases} \tag{2.2}$$

$U = A^{(n)}$ is an upper triangular matrix.

$$\ell_{ik} = \begin{cases} a_{ik}^{(k)}/a_{kk}^{(k)} & \text{if } i \geq k+1 \\ 1 & \text{if } i = k \\ 0 & \text{if } i \leq k-1. \end{cases} \tag{2.3}$$

2.3 LU decomposition w/ partial pivoting

Numerical stability and partial pivoting

If $a_{rr}^{(r)}$ is either zero or very small during the LU decomposition, the rounding error is likely to be large. One way to correct this phenomenon is to interchange the rows during the elimination so that the pivoting element is largest, called partial pivoting.

Example 2.3.1. Consider solving

$$10^{-6}x_1 + x_2 = 0.501 \tag{2.4}$$
$$x_1 - x_2 = 999.5. \tag{2.5}$$

Thr exact solution is $x_1 = 1,000, x_2 = 0.5$, with 6 digit decimal arithmetic. Eliminating x_1 from the first row, we get

$$-(10^6 + 1)x_2 = -500,000.5.$$

Since $1/(10^6 + 1) = 10^{-6}$ with 6 digit decimal arithmetic, we have $x_2 = 500,000.5 * 10^{-6} = 0.5000005 = 0.500001$ (rounded) Substituting into (2.4), we obtain

$$10^{-6}x_1 = 0.501 - 0.500001 = 0.000999$$

and hence $x_1 = 999$.

Partial pivoting

If some $a_{ii}^{(k)} = 0$ (zero pivot) or very small, we choose a j such that $a_{ji} \neq 0$(or $|a_{ji}| = \max_{j \geq i} |a_{ji}|$. Such j exists since $\det A \neq 0$.) We interchange i-th row with j-th row including the right hand side vector $\mathbf{b}$. (In the actual coding, we do not interchange rows. Instead, we merely permute the index.)

Now the pseudo algorithm for Gaussian elimination with partial pivoting is

For $k = 1, \cdots, n-1$ do
  Determine $p \geq k$ so $|a_{pk}| = \max_{i \geq k} |a_{ik}|$
  Swap a_{kj} and a_{pj} for $j = k, \cdots, n$
  for $i = k+1, \cdots, n$ do

$m_{ik} = -\frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}$;
$a_{ik}^{(k)} = -m_{ik}$ (this is to overwrite L)
for $j = k+1, \cdots, n$ do
$a_{ij}^{(k+1)} \leftarrow a_{ij}^{(k)} + m_{ik} a_{kj}^{(k)}$
end
end
end

The result of partial pivoting is of the form

$$L_{n-1} P_{n-1} \cdots L_1 P_1 A = U, \tag{2.6}$$

where P_i are permutations. Is this LU decomposition ? No! clearly

$$(L_{n-1} P_{n-1} \cdots L_1 P_1)^{-1}$$

is not a lower triangular matrix. But by multiplying $P = P_{n-1} \cdots P_1$, we can show it is an LU decomposition.

Theorem 2.3.2. *Let $P = P_{n-1} \cdots P_1$. Then we have*

$$PA = LU,$$

where

$$\begin{aligned} L &= P(L_{n-1} P_{n-1} \cdots L_1 P_1)^{-1} \\ &= P_{n-1} \cdots P_2 P_1 P_1^{-1} L_1^{-1} P_2^{-1} L_2^{-1} \cdots P_{n-1}^{-1} L_{n-1}^{-1} \\ &= (P_{n-1} \cdots P_3 (P_2 L_1^{-1} P_2^{-1}) L_2^{-1} \cdots P_{n-1}^{-1}) L_{n-1}^{-1} \end{aligned}$$

is a lower triangular matrix.

Proof. We see from (2.6) that

$$A = P_1^{-1} L_1^{-1} P_2^{-1} L_2^{-1} P_3^{-1} \cdots P_{n-1}^{-1} L_{n-1}^{-1} U.$$

Thus

$$P_2 P_1 A = P_2 L_1^{-1} P_2^{-1} L_2^{-1} P_3^{-1} \cdots P_{n-1}^{-1} L_{n-1}^{-1} U.$$

Observe that

$$L_1^{-1} = \begin{bmatrix} 1 & 0 \\ \mathbf{m}^1 & I_{n-1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \mathbf{0} \\ m_1 & 1 & \mathbf{0} \\ \mathbf{m}^* & 0 & I_{n-2} \end{bmatrix} \tag{2.7}$$

and P_2 is a permutation of rows except the first row. Hence we see $L_2' := P_2 L_1^{-1} P_2^{-1}$ has exactly the same form as L_1^{-1}.(By permuting rows and columns of $(n-1)$ identity matrix, it does not change I_{n-1}). Now

$$P_3 P_2 P_1 A = P_3 L_2' L_2^{-1} P_3^{-1} \cdots P_{n-1}^{-1} L_{n-1}^{-1} U.$$

Here $L_2' L_2^{-1}$ is the same form as (2.7). Again observe that P_3 is a permutation of rows but does not permute the first two rows. Hence $L_3' := P_3 L_2' L_2^{-1} P_3^{-1}$ is exactly the same form as $L_1^{-1} L_2^{-1}$ in step 2. Thus

$$P_3 P_2 P_1 A = L_3' L_3^{-1} P_4^{-1} L_4^{-1} \cdots P_{n-1}^{-1} L_{n-1}^{-1} U = L_3'' P_4^{-1} L_4^{-1} \cdots P_{n-1}^{-1} L_{n-1}^{-1} U.$$

Repeating this process, we see

$$L = P_{n-1} \cdots (P_2 L_1^{-1} P_2^{-1} L_2^{-1} \cdots P_{n-1}^{-1}) L_{n-1}^{-1} = P_{n-1} L_{n-2}' P_{n-1}^{-1} L_{n-1}^{-1} = L_{n-1}' L_{n-1}^{-1}.$$

Here the product of last matrices is a lower triangular matrix. □

pseudo code

input $n, (a_{ij})$
for $k = 1, 2, \cdots, n$ do
  $p_i \leftarrow i$
end do
for $k = 1, 2, \cdots, n-1$ do
  select $j \geq k$ so that
    $|a_{p_j,k}| \geq |a_{p_i,k}|$ for $i = k, k+1, \cdots, n$
  $p_k \leftrightarrow p_j$
  for $i = k+1, k+2, \cdots, n$ do
    $m \leftarrow a_{p_i,k}/a_{p_k,k}$; $a_{p_i,k} \leftarrow m$(saving multiplier in lower diag)
    for$j = k+1, k+2, \cdots, n$ do
      $a_{p_i,j} \leftarrow a_{p_i,j} - m a_{p_k,j}$ —- (*)
    end
  end

end
output $(a_{ij}), (p_i)$

Remark 2.3.3. We do not actually move the rows of A. We merely change the order of elimination process according to the pivot row.

2.3.1 Keeping track of Permutation

Note $P = P_{n-1} \cdots P_1$ is the product of permutation matrices involved in the pivoting process. We do not save all the $P_i's$. Instead, we star with a pivot vector $\mathbf{p} = [1, 2, 3, \cdots, n]$ and permute it according to P.(The i-th component p_i denotes the i-th pivot row) Since the system $A\mathbf{x} = \mathbf{b}$ is changed to $PA\mathbf{x} = P\mathbf{b}$, we need to permute $\mathbf{b}$ accordingly to form $P\mathbf{b}$.

Example 2.3.4. If the final $\mathbf{p}$ is $[2, 4, 3, 1]$, then we see that the rows of PA are the rows of A in the order of $2, 4, 3, 1$. So the right hand side $\mathbf{b}$ can be permuted in the same way, i.e., $P\mathbf{p} = (b_2, b_4, b_3, b_1)$.

2.3.2 Back -substitution

From

$$A\mathbf{x} = \mathbf{b}$$

$$PA\mathbf{x} = P\mathbf{b} \Rightarrow LU\mathbf{x} = \mathbf{d}.$$

Solving consists of two steps :

$$L\mathbf{y} = \mathbf{d} \text{ and } U\mathbf{x} = \mathbf{y}.$$

First step is to solve for $\mathbf{y}$ from $L\mathbf{y} = \mathbf{d}$.(Elimination of right hand side — this could have been done during the Elimination step, (*) above)

input $(a_{ij}), (p_i), (b_i)$
for $k = 1, 2, \cdots, n-1$ do
  for $i = k+1, \cdots, n$ do
    $b_{p_i} \leftarrow b_{p_i} - m_{p_i,k} b_{p_k} = b_{p_i} - a_{p_i,k} b_{p_k}$;
  end
end

Now get $\mathbf{x}$ from $U\mathbf{x} = \mathbf{y}$(Backward substitution)

for $i = n, n-1, \cdots, 1$ do
  for $j = i+1, \cdots, n$ do
    $temp = b_{p_i} - m_{p_i,j} x_j$;
    $x_i \leftarrow temp/a_{p_i,i}$
  end
end output (x_i)

Storage

To save the storage, we use the memory that was used to save A to record L and U(i.e, when A is not needed anymore, we overwrite it.) Since we do not need to store 1's on the diagonal, we store

$$A \to [L\backslash U] = \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1,n-1} & u_{1n} \\ m_{21} & u_{22} & \cdots & u_{2,n-1} & u_{2n} \\ m_{31} & m_{32} & \ddots & u_{3,n-1} & u_{3n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ . & ... & \cdots & u_{n-1,n-1} & . \\ m_{n1} & m_{n2} & \cdots & m_{n,n-1} & u_{nn} \end{bmatrix}$$

2.4 Cholesky Decomposition

When we know some special property of a matrix, we often would like to exploit it. Those are typically

(1) Sparseness

(2) Symmetry

(3) Positive definiteness

A symmetric matrix A is said to be **positive definite**[1] if for any nonzero vector $\mathbf{x}$

$$\mathbf{x}^T A\mathbf{x} > 0.$$

The quantity $\mathbf{x}^T A\mathbf{x} > 0$ is called a **quadratic form**. A matrix is positive definite if and only if it is symmetric and all the eigenvalues are positive.

For a symmetric matrix, the storage is about the half of general matrix.

[1]Most authors do not include the symmetry in the definition of positive definite matrix

In the case of LU decomposition $A = LU$ of a symmetric matrix, L, U are not symmetric. Even we do not have $L^T = U$, since L is unit lower triangular. How can we make the decomposition as symmetric as possible so that we can at least save some storage and efficiently apply the algorithm?

Theorem 2.4.1. *If A is real, symmetric and positive definite matrix, then it has a unique factorization $A = L_c L_c^T$ where L is a lower triangular matrix with positive diagonal.*

It is called **Cholesky decomposition** or **Cholesky factorization**.

Let us see how to accomplish it. Since A is symmetric, we have an orthogonal diagonalization

$$A = SDS^T,$$

where columns of S are orthonormal eigenvectors and $D = diag(\lambda_1, \cdots, \lambda_n)$. Since $\lambda_i > 0$, we may define $H = diag(\sqrt{\lambda_1}, \cdots, \sqrt{\lambda_n})$ and

$$\begin{aligned} A &= SHH^TS^T \\ &= (SH)(SH)^T \\ &= MM^T. \end{aligned}$$

Certainly M is not a lower triangular matrix. It is more useful to write it in the form

$$\begin{aligned} A &= SHH^TS^T \\ &= (SH)(S^TS)HS^T \\ &= (SHS^T)(SHS^T) \\ &= H_0^2. \end{aligned}$$

$H_0 = SHS^T$ is called the **square root** of A. H_0 is SPD. The matrix square root appears in many physical applications, and we use this to derive Cholesky decomposition.

Let E be the elementary row operation involved in LU decomposition. Then $EAE^T = EH_0H_0^TE^T$. Repeating the process as in step (1), we obtain

$$E_{n1} \cdots E_{31}E_{21}A = L_1A.$$

Here multiplying the matrices $E_{n1}\cdots E_{31}E_{21}$ on the left has the effect of row reducing. Since A is symmetric, multiplying $E_{21}^T E_{31}^T \cdots E_{n1}^T$ on the right will have similar effect(column reducing effect) on the columns of A.

$$E_{n1}\cdots E_{31}E_{21}AE_{21}^T E_{31}^T \cdots E_{n1}^T = L_1 A L_1^T$$

will have the following (symmetric) form

$$\begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} \\ 0 & \vdots & \vdots & \vdots \\ 0 & a_{n2}^{(2)} & \cdots & a_{nn}^{(2)} \end{bmatrix}$$

Similarly, we repeat the same process to obtain a diagonal matrix:

$$L_{n-1}\cdots L_2 L_1 A L_1^T L_2^T \cdots L_{n-1}^T = D_0.$$

Let $H_0 = \sqrt{D_0}$ which is again a diagonal matrix. Then with $\tilde{L} = L_{n-1}\cdots L_2 L_1$ we have $\tilde{L}A\tilde{L}^T = D_0$ and with $H_0 := \sqrt{D_0}$

$$\begin{aligned} A &= \tilde{L}^{-1}D_0(\tilde{L}^T)^{-1} \\ &= \tilde{L}^{-1}H_0^2(\tilde{L}^{-1})^T \\ &= (\tilde{L}^{-1}H_0)(\tilde{L}^{-1}H_0)^T \\ &= L_c L_c^T. \end{aligned}$$

Since $\tilde{L}^{-1}$ is the unit lower triangular L in (2.1), we have

$$U = H_0(H_0\tilde{L}^{-T}) = H_0 H_0 L^T. \tag{2.8}$$

Since L is a unit lower triangular matrix and H_0 is diagonal, we see that H_0^2 is the diagonal of U. One can show the decomposition is unique. In fact by uniqueness,

$$A = LU = LD\tilde{U} = L\sqrt{D}\sqrt{D}\tilde{U} = L_c L_c^T. \tag{2.9}$$

Thus the Cholesky decomposition is obtained from LU-decomposition by taking square root of diagonals of U.

2.5 Condition Numbers

Vector Norms

Definition 2.5.1. A vector norm $\|\cdot\| : \mathbb{C}^n \to \mathbb{R}^1$ is a function satisfying

(1) $\|\mathbf{x}\| \geq 0$ and $\|\mathbf{x}\| = 0$ iff $\mathbf{x} = 0$

(2) $\|\alpha\mathbf{x}\| = |\alpha| \, \|\mathbf{x}\|$ for all scalar α

(3) $\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$

Example 2.5.2. (1) $\|\mathbf{x}\|_1 = \sum |x_i|$

(2) $\|\mathbf{x}\|_p = (\sum |x_i|^p)^{\frac{1}{p}}$

(3) $\|\mathbf{x}\|_2 = (\sum |x_i|^2)^{\frac{1}{2}} = (\mathbf{x}^H \cdot \mathbf{x})^{\frac{1}{2}} \equiv (\mathbf{x}, \mathbf{x})^{\frac{1}{2}}$

(4) $\|\mathbf{x}\|_\infty = \max_i |x_i|$

Theorem 2.5.3. *We have*

(1) $\|\mathbf{x}\|_\infty \leq \|\mathbf{x}\|_2 \leq \|\mathbf{x}\|_1 \leq n\|\mathbf{x}\|_\infty$

(2) $\|\mathbf{x}\|_2 \leq n^{\frac{1}{2}}\|\mathbf{x}\|_\infty$

(3) $n^{-\frac{1}{2}}\|\mathbf{x}\|_1 \leq \|\mathbf{x}\|_2$.

Definition 2.5.4. We say $f : \mathbb{C}^n \to \mathbb{R}$ is uniformly continuous, if for each $\mathbf{x} \in \mathbb{C}^n$, and given $\varepsilon > 0$ there exists a $\delta > 0$ such that $|f(\mathbf{x}) - f(\mathbf{y})| < \varepsilon$, whenever $\|\mathbf{x} - \mathbf{y}\|_\infty < \delta$.

Theorem 2.5.5. *If $\|\cdot\|$ is any norm, then it is a continuous function.*

Proof.

$$\begin{aligned} |\,\|\mathbf{x}\| - \|\mathbf{y}\|\,| &\leq \|\mathbf{x} - \mathbf{y}\| = \left\| \sum_{i=1}^n (x_i - y_i) e_i \right\| \leq \max |x_i - y_i| \sum_{i=1}^n \|e_i\| \\ &= \|\mathbf{x} - \mathbf{y}\|_\infty \sum_{i=1}^n \|e_i\|. \end{aligned}$$

Thus if we choose $\delta < \frac{\epsilon}{\sum_{i=1}^n \|e_i\|}$, then we see

$$|\,\|\mathbf{x}\| - \|\mathbf{y}\|\,| < \epsilon.$$

□

Definition 2.5.6. We say two norms $\|\cdot\|_\alpha$ and $\|\cdot\|_\beta$ are equivalent if there exist positive constants C_0, C_1 such that

$$C_0\|\mathbf{x}\|_\alpha \le \|\mathbf{x}\|_\beta \le C_1\|\mathbf{x}\|_\alpha, \quad \text{for all } \mathbf{x} \in V.$$

Theorem 2.5.7. *In a finite dimensional vector space V, all norms are equivalent. In other words, if $\|\cdot\|_\alpha$ and $\|\cdot\|_\beta$ are any two norms, then there exist positive constants m, M such that $C_0\|\mathbf{x}\|_\alpha \le \|\mathbf{x}\|_\beta \le C_1\|\mathbf{x}\|_\alpha$ for all $\mathbf{x} \in V$.*

Proof. We shall show every norm is equivalent to $\|\cdot\|_\infty$. First let $S = \{\mathbf{y} \in V : \|\mathbf{y}\|_\infty = 1\}$. Then S is closed and bounded. Let $\|\cdot\|$ be any other norm. Since any norm is a continuous function, it assumes a positive max and min on S, i.e., $0 < m \le \|\mathbf{y}\| \le M$ on S. Then for any $\mathbf{x} \neq 0 \in \mathbb{C}^n$, $\frac{\mathbf{x}}{\|\mathbf{x}\|_\infty} \in S$. Hence $m \le \left\|\frac{\mathbf{x}}{\|\mathbf{x}\|_\infty}\right\| \le M$. In other words,

$$m\|\mathbf{x}\|_\infty \le \|\mathbf{x}\| \le M\|\mathbf{x}\|_\infty.$$

This inequality obviously holds for $\mathbf{x} = 0$. Finally transitivity of equivalence relation proves the result. □

Matrix Norms

Definition 2.5.8. A real-valued function $\|\cdot\|$ defined for all $n \times n$ matrices is said to be a **matrix norm** if it satisfies

(1) $\|A\| \ge 0$ and $\|A\| = 0$ iff $A \equiv 0$

(2) $\|cA\| = |c|\,\|A\|$

(3) $\|A + B\| \le \|A\| + \|B\|$

(4) $\|A \cdot B\| \le \|A\| \cdot \|B\|$

(5) $\|A\mathbf{x}\| \le \|A\| \cdot \|\mathbf{x}\|$

If $\|\cdot\|_v$ is a vector norm on $\mathbb{R}^n$ and $\|\cdot\|_M$ is a matrix norm on $n \times n$ matrices, then we say they are **consistent** (or **compatible**) if

$$\|A\mathbf{x}\|_v \le \|A\|_M\|\mathbf{x}\|_v.$$

Example 2.5.9. (1) A natural way to consider a norm on $n \times n$ matrices is to treat the set of $n \times n$ matrices as a vector space $\mathbb{R}^{n \times n}$ and consider Euclidean norm:

$$\|A\|_F = \sqrt{\sum_{i,j} a_{ij}^2}.$$

This is called a **Frobenius norm** It can be shown that

$$\|A\|_F = \sqrt{tr(A^T A)},$$

where $tr(M)$ is the trace of M, the sum of diagonal entries of M.

(2) However, the most important norm is the **spectral norm**:

$$\|A\|_s = \sqrt{\rho(A^T A)}$$

where $\rho(M)$ is the spectral radius of M, that is the largest eigenvalue of M.

Let $\|\cdot\|_v$ be a vector norm. Then the norm $\|A\|_M$ of A defined by

$$\|A\|_M := \sup_{\mathbf{x} \neq 0} \frac{\|A\mathbf{x}\|_v}{\|\mathbf{x}\|_v} \equiv \sup_{\|\mathbf{x}\|_v = 1} \|A\mathbf{x}\|_v$$

is called the **induced**, or **natural norm**. An induced norm is always consistent with the vector norm that induces it. In other words

$$\|A\mathbf{x}\|_v \leq \|A\|_M \|\mathbf{x}\|_v.$$

In addition, if $\|\cdot\|_\mu$ is any other matrix norm consistent with $\|\cdot\|_v$ then for every $n \times n$ matrix A,

$$\rho(A) \leq \|A\|_M \leq \|A\|_\mu. \tag{2.10}$$

Prove it. Thus $\rho(A)$ is a lower bound. One can show that the spectral norm is the norm induced by Euclidean norm. We also have

$$\|A\|_M \leq \|A\|_F. \tag{2.11}$$

Frobenius norm is easier to compute

Theorem 2.5.10. *Let $A = [a_{ij}] \in \mathbb{C}^{n,n}$. Then we have*

(1) $\|A\|_1 = \max_j \sum_i |a_{ij}|$*,(maximum column sum),*

(2) $\|A\|_2 = \sqrt{\rho(A^H A)}$*,* $\|A\|_2 = \rho(A)$ *if* $A = A^H$*,*

(3) $\|A\|_\infty = \max_i \sum_j |a_{ij}|$, *(maximum row sum).*

Condition Number

Let us apply the matrix norm to see how the property of A affect the computed solution $\hat{\mathbf{x}}$ of $A\mathbf{x} = \mathbf{b}$. We will measure the relative error

$$\frac{\|\mathbf{x} - \hat{\mathbf{x}}\|}{\|\mathbf{x}\|}.$$

Consider the residual error

$$\begin{aligned} \mathbf{r} &= \mathbf{b} - A\hat{\mathbf{x}} \\ &= A\mathbf{x} - A\hat{\mathbf{x}} \\ &= A(\mathbf{x} - \hat{\mathbf{x}}). \end{aligned}$$

Take the norms

$$\begin{aligned} \|\mathbf{r}\| &= \|A(\mathbf{x} - \hat{\mathbf{x}})\| \\ &\leq \|A\|\|\mathbf{x} - \hat{\mathbf{x}}\|. \end{aligned}$$

This does not help since we want to estimate $\|\mathbf{x} - \hat{\mathbf{x}}\|$. So let go the other way.

$$\begin{aligned} \mathbf{r} &= \mathbf{b} - A\hat{\mathbf{x}} \\ &= A\mathbf{x} - A\hat{\mathbf{x}} \\ A^{-1}\mathbf{r} &= \mathbf{x} - \hat{\mathbf{x}} \\ \|\mathbf{x} - \hat{\mathbf{x}}\| &\leq \|A^{-1}\|\|\mathbf{r}\|. \end{aligned}$$

Divide by $\|\mathbf{x}\|$

$$\frac{\|\mathbf{x} - \hat{\mathbf{x}}\|}{\|\mathbf{x}\|} \leq \frac{\|A^{-1}\|\|\mathbf{r}\|}{\|\mathbf{x}\|}. \tag{2.12}$$

On the left, we have a quantity we want to estimate, while on the right we need to replace it by something we can estimate(the relative error of input data). From $A\mathbf{x} = \mathbf{b}$, we have

$$\begin{aligned} \|\mathbf{b}\| &\leq \|A\|\|\mathbf{x}\| \\ \frac{\|\mathbf{b}\|}{\|A\|} &\leq \|\mathbf{x}\|. \end{aligned}$$

Underestimating $\|\mathbf{x}\|$ from below, we have from (2.12)

$$\frac{\|\mathbf{x}-\hat{\mathbf{x}}\|}{\|\mathbf{x}\|} \leq \frac{\|A^{-1}\|\|\mathbf{r}\|}{\|\mathbf{x}\|} \leq \frac{\|A^{-1}\|\|\mathbf{r}\|}{\left(\frac{\|\mathbf{b}\|}{\|A\|}\right)}. \tag{2.13}$$

Thus

$$\frac{\|\mathbf{x}-\hat{\mathbf{x}}\|}{\|\mathbf{x}\|} \leq \|A\|\|A^{-1}\|\frac{\|\mathbf{r}\|}{\|\mathbf{b}\|}. \tag{2.14}$$

We define the **condition number** of A by

$$\kappa(A) = \|A\|\|A^{-1}\|.$$

The condition number depends on the particular norm we choose. The condition number is always greater than or equal to 1.(Show it)

This estimate is often (almost) sharp. i.e, if $\kappa(A) \approx 1$ the error is small but if $\kappa(A) >> 1$ the error is usually large.

Example 2.5.11. The solution of $A\mathbf{x} = \mathbf{b}$ for certain matrix A depending on ϵ and a unit vector $\mathbf{b}$, we have

$$\mathbf{x} = \frac{1}{\epsilon^2}\begin{pmatrix}1\\1\end{pmatrix}$$

Assume 16 digit machine. With $\epsilon = 0.1$, $\kappa(A) \approx 10^4$ and the exact solution x_1 and approximate solution $\hat{x}_1$ are

$$x_1 = 70.71067811865474, \hat{x}_1 = 70.71067811864299.$$

The accurate up to 12 digits(70.7106781186) and the error is

$$x_1 - \hat{x}_1 \approx 1.175 \times 10^{-11}, \quad \frac{x_1 - \hat{x}_1}{x_1} \approx 1.66 \times 10^{-13}$$

Now with $\epsilon = 0.01$, $\kappa(A) \approx 10^8$ and we have

$$x_1 = 7071.067811865475, \hat{x}_1 = 7071.06791978500.$$

$$x_1 - \hat{x}_1 \approx 7.2 \times 10^{-4}, \quad \frac{x_1 - \hat{x}_1}{x_1} \approx 2.16 \times 10^{-8}.$$

The accuracy of x is only seven digits (7071.067). We say this is sensitive.

Another view is this: we usually are apt to commit some errors in the A and $\mathbf{b}$ so that we are solving

$$A_1\xi = \mathbf{b}_1.$$

Then the error is

$$\frac{\|\xi - \mathbf{x}\|}{\|\mathbf{x}\|} \approx \kappa(A)\epsilon.$$

Again the error seems to be amplified by the amount of the condition number.

Example 2.5.12. The exact solution of the system

$$\begin{array}{rrcr} 3x+ & 5y & = & 8 \\ 2x+ & 3.3333y & = & 5.3333 \end{array}$$

is $(1, 1)$. We use Gaussian elimination to solve it. With a calculator having 5 significant digits, we have(by rounding) $2/3 = .66667$. Thus the multiplier $m_{21} = .6666\dot{6} \approx 0.66667 \times 5 = 3.333335$ and $.66667 \times 8 = 5.33336$. Hence eliminating x we obtain

$$-0.000035y = -0.00006$$

which gives $y = 1.7$.

Example 2.5.13. The exact solution of the system

$$\begin{array}{rrcr} 3x+ & 5y & = & 8 \\ 2x+ & 3.3333333y & = & 5.3333333 \end{array}$$

is $(1, 1)$. With a calculator having 10 significant digits, we have(by chopping) $2/3. = 0.666666666(9\ digits)$.

Thus the coefficient of y in the second row is

$$3.3333333 - 5m_{21} = 3.3333333 - 0.666666666 \times 5 = -0.00000003$$

and $0.666666666 \times 8 = 5.333333328$. Hence eliminating x

$$\begin{array}{rrcr} 3x+ & 5y & = & 8 \\ & -0.00000003y & = & -0.000000028 \end{array}$$

which gives $y = 0.93333333$. What happened ? Now let us change the order of elimination. Multiplying the second equation by $3/2$, the exact equation

after one step of elimination is

$$\begin{array}{rrcr} 3x+ & 5y & = & 8 \\ 3x+ & 4.99999995y & = & 7.99999995 \end{array}$$

Thus eliminating x, we have

$$-0.00000005y = -0.00000005.$$

Hence $y = 1$. Thus we see the result depends on the method of elimination.

Ill-conditioned system: Least square approximation

Given $f(x) \in L^2(0,1)$, find a polynomial of degree n such that

$$\int_0^1 (f(x) - p_n(x))^2 dx$$

is minimum. Let $p_n(x) = \sum_{i=0}^n c_i x^i$. Then the minimum is attained when

$$\frac{\partial}{\partial c_j} \int_0^1 (f(x) - p_n(x))^2 dx = 0, \quad \text{for } j = 0, 1, \cdots, n.$$

Hence, we get

$$\sum_{i=0}^n c_i \int_0^1 x^i x^j \, dx = \int_0^1 f(x) x^j \, dx, \quad j = 0, 1, \cdots, n$$

which can be written as

$$A\mathbf{c} = \mathbf{f},$$

where $A_{ij} = \frac{1}{i+j+1}$, $i, j = 0, \cdots, n$, $\mathbf{c} = (c_0, \cdots, c_n)$ and

$$\mathbf{f} = \left(\int_0^1 f(x) x^0 dx, \cdots, \int_0^1 f(x) x^n dx \right).$$

This matrix is called Hilbert matrix and is extremely ill-conditioned! If we choose $f(x) = Q_n(x) = \sum_{i=0}^n q_i x^i$, then $c_i = q_i$, so that one can compare the numerical solution.

Example 2.5.14. If $f(x) = \sum_{i=0}^n x^i$, $\mathbf{f} = (\sum_{i=0}^n \frac{1}{i+j+1})_{j=0}^n$. Compute $\mathbf{c}$ and compare with exact solution $(1, \cdots, 1)$.

Exercise 2.5.15. (1) Consider solving

$$-u'' = f \text{ on } (0,1), \text{ with B.C } u(0) = a, \quad u(1) = b \tag{2.15}$$

by finite difference method or finite element method. First subdivide the interval by n-equal intervals of length $h = 1/n$.

$$x_0 = 0, x_1 = h, \cdots, x_i = ih, \cdots, x_n = 1.$$

We shall get

$$Au_h = f_h,$$

where A_h is $(n-1) \times (n-1)$ system given by

$$A_h = \frac{1}{h^2} \begin{bmatrix} 2 & -1 & & \\ -1 & 2 & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{bmatrix}$$

and

$$f_h = \left(f(x_1) + \frac{u(0)}{h^2}, f(x_2), \cdots, f(x_{n-1}) + \frac{u(1)}{h^2} \right)^T.$$

When $f = x(x-1)$, the exact solution is $u = -\frac{x^4}{12} + \frac{x^3}{6} - \frac{x}{12}$. Solve the BVP (2.15) with $n = 10, 20, 30, 40,$ etc. and compare with exact solution(Check $\|u_h - u\| := \sqrt{\sum_i ((u_h - u)(x_i))^2 h}$— This is a kind of discrete L^2-norm) and find a pattern how the error is decreasing.

Eigenvalues of $(n-1) \times (n-1)$ matrix $h^2 A$ are $\lambda_j = 2(1 - \cos j\pi h)$, and eigenvectors are $\mathbf{x}_j = (\sin j\pi h, \cdots, \sin(n-1)j\pi h)$. For $n = 10$, the eigenvalues are

$$3.90211, 3.61803, 3.1755, 2.618, 1.9999, 1.3819, 0.8244, 0.38196, 0.9788$$

Solve

Example 2.5.16 (2 D)**.** A typical 2nd order elliptic differential equation is

$$\begin{aligned} -\Delta u &= f \text{ on } \Omega \\ u &= g \text{ on } \partial\Omega \end{aligned}$$

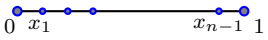

Figure 2.1: meshes for boundary value problem

where f and g are given. We take $\Omega = [0,1]^2$. With $h = 1/n$, the derivatives are approximated by

$$\begin{aligned} u_{xx}(x,y) &\doteq [u(x+h,y) - 2u(x,y) + u(x-h,y)]/h^2 \\ u_{yy}(x,y) &\doteq [u(x,y+h) - 2u(x,y) + u(x,y-h)]/h^2. \end{aligned}$$

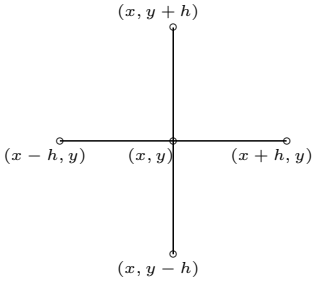

Figure 2.2: 5-point Stencil

The figure is called a **Molecule, Stencil**. For each point (interior mesh pt), approximate $\nabla^2 u = \Delta u$ by 5-point stencil. By Gershgorin disc theorem, the matrix is nonsingular.

When the unit square is divided by $n = 1/h$ equal intervals along x-axis and y-axis, then the corresponding matrix A is $(n-1)\times(n-1)$ block-diagonal matrix of the form:

$$A = \frac{1}{h^2}\begin{bmatrix} B & -I & 0 & \cdots & \\ -I & B & -I & 0 & \\ & -I & \ddots & \ddots & \\ & & \ddots & B & -I \\ & \cdots & 0 & -I & B \end{bmatrix} \tag{2.16}$$

where

$$B = \begin{bmatrix} 4 & -1 & 0 & \cdots & \\ -1 & 4 & -1 & 0 & \\ & -1 & \ddots & \ddots & \\ & & \ddots & 4 & -1 \\ & \cdots & 0 & -1 & 4 \end{bmatrix}$$

is $(n-1) \times (n-1)$ matrix. The eigenvectors of $(n-1)^2 \times (n-1)^2$ matrix $h^2 A$ are

$$x^h_{\nu\mu}(x_i, y_j) = \sin(\nu\pi x_i)\sin(\mu\pi y_j), \quad (x_i, y_j) = (hi, hj) \in \Omega_h \tag{2.17}$$

with the corresponding eigenvalues

$$\lambda^h_{\nu\mu} = 4h^{-2}(\sin^2(\nu\pi h/2) + \sin^2(\mu\pi h/2)), \quad 1 \le \nu, \mu \le n-1. \tag{2.18}$$

Note that the eigenvalues of Laplace equation for the domain $[0, a] \times [0, b]$ are

$$\lambda_{\mu\nu} = (\frac{\mu^2}{a^2} + \frac{\nu^2}{b^2})\pi^2, \ \mu, \nu = 1, \cdots, . \tag{2.19}$$

2.6 The QR decomposition

When A is $m \times n$ matrix ($m \ge n$), we can still consider a solution of $A\mathbf{x} = \mathbf{b}$. The following QR decomposition is possible:

$$A = QR$$

where Q is $m \times m$ orthogonal matrix($Q^T = Q^{-1}$) and R is an $m \times n$ upper triangular matrix. For example if A is 5×3, then

$$A = \begin{bmatrix} X & X & X \\ X & X & X \\ X & X & X \\ X & X & X \\ X & X & X \end{bmatrix} = \begin{bmatrix} X & X & X & X & X \\ X & X & X & X & X \\ X & X & X & X & X \\ X & X & X & X & X \\ X & X & X & X & X \end{bmatrix} \begin{bmatrix} X & X & X \\ 0 & X & X \\ 0 & 0 & X \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Suppose A is a square and have a QR decomposition, then we can solve $A\mathbf{x} = \mathbf{b}$:

$$\begin{aligned} A\mathbf{x} &= \mathbf{b} \\ QR\mathbf{x} &= \mathbf{b} \\ R\mathbf{x} &= Q^T\mathbf{b} \end{aligned}$$

Since R is upper triangular, this is easy to solve provided that diagonal entries of R are nonzero. Note that

$$\begin{aligned} det(A) &= det(QR) \\ &= det(Q)det(R) \\ &= \pm det(R) \end{aligned}$$

Thus $det(A) \neq 0$ if and only $det(R) \neq 0$. Since R is an upper triangular matrix, its determinant is product of diagonal entries.

Now consider the case of **nonsquare** matrix. The case $m > n$(more equations than the unknowns) is more interesting. Since the product $A\mathbf{x}$ is a linear combination of columns of A, the system $A\mathbf{x}$ always has a solution when $\mathbf{b}$ can be written as a linear combinations of columns of A or

$$\mathbf{b} \in Col(A)(\text{ consistency })$$

or $\mathbf{b}$ must be in the range of A.

What can we do if the system is inconsistent?

Least Square Methods

Consider the following data:

$$(x_1, f(x_1)), \cdots, (x_n, f(x_n)).$$

What is the closest linear function to f?

The idea is to find a linear function $L(x) = mx + d$ such that

$$\|L(\mathbf{x}) - f(\mathbf{x})\|$$

is minimized! Here $\mathbf{x} = (1, 2, 3, 4, 5)$. But in what sense?

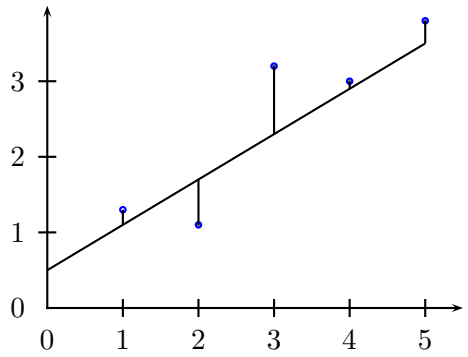

Figure 2.3: fitting by linear function using least square method

Let use use L^2-norm. Thus we want to minimize

$$E = \sqrt{\sum_{i=1}^{5} |L(x_i) - f(x_i)|^2} \text{ or } \sum_{i=1}^{5} |L(x_i) - f(x_i)|^2.$$

Written as linear system in (m, d) -unknowns, we have

$$\min_{m,d} \left\| A \begin{pmatrix} m \\ d \end{pmatrix} - f(\mathbf{x}) \right\|$$

where

$$A = \begin{pmatrix} x_1 & 1 \\ x_2 & 1 \\ \cdots & \\ x_5 & 1 \end{pmatrix}, f(\mathbf{x}) = \begin{pmatrix} f(x_1) \\ f(x_2) \\ \cdots \\ f(x_5) \end{pmatrix}.$$

To get the minimum, we take derivative with respect to m and b. ($\frac{\partial E}{\partial m} = 0, \frac{\partial E}{\partial d} = 0.$) Thus

$$\begin{array}{rcl} 2\sum_{i=1}^{5}(mx_i + d - f(x_i))x_i & = & 0 \\ 2\sum_{i=1}^{5}(mx_i + d - f(x_i)) & = & 0. \end{array} \Longrightarrow \begin{array}{rcl} m\sum_{i=1}^{5} x_i^2 + d\sum_{i=1}^{5} x_i & = & \sum_{i=1}^{5} f(x_i)x_i \\ m\sum_{i=1}^{5} x_i + 5d & = & \sum_{i=1}^{5} f(x_i) \end{array}$$

In its least square equation, we have $A^TA\mathbf{m} = A^T\mathbf{f}$. In its entry

$$\begin{pmatrix} x_1 & x_2 & \cdots & x_5 \\ 1 & 1 & \cdots & 1 \end{pmatrix} \begin{pmatrix} x_1 & 1 \\ x_2 & 1 \\ \cdots & \\ x_5 & 1 \end{pmatrix} \begin{pmatrix} m \\ d \end{pmatrix} = \begin{pmatrix} x_1 & x_2 & \cdots & x_5 \\ 1 & 1 & \cdots & 1 \end{pmatrix} \begin{pmatrix} f(x_1) \\ f(x_2) \\ \cdots \\ f(x_5) \end{pmatrix}$$

Changing to ordinary notation, we want to find minimizer $\mathbf{x}$ of

$$\|A\mathbf{x} - \mathbf{b}\|$$

which is equivalent to minimizing

$$(A\mathbf{x} - \mathbf{b})^T(A\mathbf{x} - \mathbf{b}) = \mathbf{x}^TA^TA\mathbf{x} - \mathbf{b}^TA\mathbf{x} - \mathbf{x}^TA^T\mathbf{b} + \mathbf{b}^T\mathbf{b}.$$

Taking the derivative w.r.t $\mathbf{x}$, we have

$$\mathbf{x}^TA^TA - \mathbf{b}^TA = 0 \text{ or } A^TA\mathbf{x}^T = A^T\mathbf{b}.$$

Another interpretation: Since $A\mathbf{x}$ lies in $\mathbb{R}^n$(represented by the x-axis in the

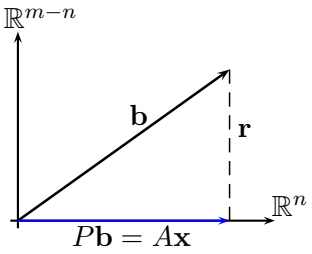

Figure 2.4: least square fit

figure), the minimum is attained when $A\mathbf{x}$ is the projection of $\mathbf{b}$ on the $\mathbb{R}^n$ space. Thus $\mathbf{r}$ is orthogonal to $\mathbb{R}^n$(the column space of A). In algebraic notation, we must have

$$\mathbf{r}^TA = 0.$$

$$\begin{aligned} A^T\mathbf{r} &= 0 \\ A^T(\mathbf{b} - A\mathbf{x}) &= 0 \\ A^TA\mathbf{x} &= A^T\mathbf{b}. \end{aligned}$$

This is called the **normal equation**. From now on we assume $m \geq n$ and the rank of A is n. Solving such problems, one could consider using LU decomposition, in this case Cholesky decomposition. However, experience says it is expansive and very sensitive.

Solving with QR

Consider solving $A^T A\mathbf{x} = A^T\mathbf{b}$. If we have a QR decomposition,

$$\begin{aligned} (QR)^T(QR)\mathbf{x} &= (QR)^T\mathbf{b} \\ R^TQ^T(QR)\mathbf{x} &= R^TQ^T\mathbf{b} \\ R^TR\mathbf{x} &= R^TQ^T\mathbf{b}. \end{aligned}$$

Can we solve this ? Note R is $m \times n$ upper triangular matrix. Thus

$$R = \begin{pmatrix} R_1 \\ \mathbf{0} \end{pmatrix}$$

where R_1 is $n \times n$ upper triangular matrix and $\mathbf{0}$ is $(m-n) \times n$ zero matrix.

$$\begin{aligned} \begin{pmatrix} R_1 \\ \mathbf{0} \end{pmatrix}^T \begin{pmatrix} R_1 \\ \mathbf{0} \end{pmatrix} \mathbf{x} &= \begin{pmatrix} R_1 \\ \mathbf{0} \end{pmatrix}^T Q^T\mathbf{b} \\ \begin{pmatrix} R_1^T & \mathbf{0}^T \end{pmatrix} \begin{pmatrix} R_1 \\ \mathbf{0} \end{pmatrix} \mathbf{x} &= \begin{pmatrix} R_1^T & \mathbf{0}^T \end{pmatrix} Q^T\mathbf{b} \\ R_1^TR_1\mathbf{x} &= \begin{pmatrix} R_1^T & \mathbf{0}^T \end{pmatrix} Q^T\mathbf{b}. \end{aligned}$$

Since R_1 is nonsingular, we have

$$R_1\mathbf{x} = \begin{pmatrix} I_n & \mathbf{0}^T \end{pmatrix} Q^T\mathbf{b},$$

that is,

$$R_1\mathbf{x} = (Q^T\mathbf{b})_n.(\text{first } n \text{ components of } Q^T\mathbf{b}) \tag{2.20}$$

This is a square system with nonsingular upper $n \times n$ triangular matrix R_1. This can be solved by backward substitution.

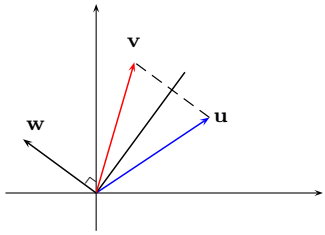

Figure 2.5: Action of an elementary reflector $H_{\mathbf{w}}$

Uniqueness of QR

In general, QR decomposition is not unique, as we can see with $QR = (-Q)(-R)$. But if we assume(we usually do) the diagonal entries of R are positive, the QR decomposition is unique.

2.7 Householder Triangularization and the QR decomposition

Let use **Householder reflection, elementary reflector** which is given by

$$H_w = I - 2\mathbf{w}\mathbf{w}^T$$

for some unit vector $\mathbf{w} \in \mathbb{R}^n$. They are symmetric and orthogonal($\mathbf{w}^T = \mathbf{w}$ and $\mathbf{w}^T\mathbf{w} = I$).

The idea is : Given $\mathbf{u}$, $\mathbf{v}$ find a $\mathbf{w}$ so that $H_{\mathbf{w}}\mathbf{u} = \mathbf{v}$. Thus from the figure we can see

$$\mathbf{w} = \frac{\mathbf{v} - \mathbf{u}}{\|\mathbf{v} - \mathbf{u}\|}. \tag{2.21}$$

Now the first step of QR is to reduce the first column to a multiple of $\mathbf{e}_1$: Let $A = [\mathbf{a}_1, A_{12}]$. Then we can find an orthogonal matrix $Q_1 = H_{\mathbf{w}}$, which maps the first column of A onto $\alpha\mathbf{e}_1$. Since $\|H_{\mathbf{w}}\mathbf{a}_1\| = \|\mathbf{a}_1\| = \|\alpha\mathbf{e}_1\|$, we have $\alpha = \pm\|\mathbf{a}_1\|$. Thus $\mathbf{w}$ is given by

$$\mathbf{w} = \frac{\alpha\mathbf{e}_1 - \mathbf{a}_1}{\|\alpha\mathbf{e}_1 - \mathbf{a}_1\|}. \tag{2.22}$$

and

$$Q_1 A = \begin{pmatrix} \alpha & b_2 & \dots & b_n \\ 0 & & & \\ \vdots & & A_{12} & \\ 0 & & & \end{pmatrix}$$

We repeat the same process. If $(n-1)\times(n-1)$ matrix Q_2' is the elementary reflector used to reduce the first column of A_{12}, then the matrix

$$Q_2 = \begin{pmatrix} 1 & 0 \\ 0 & Q_2' \end{pmatrix}$$

makes

$$Q_2 Q_1 A = \begin{pmatrix} \alpha & b_2 & \dots & b_n \\ 0 & \alpha_2 & * & * \\ \vdots & 0 & A_{23}' & \\ 0 & 0 & & \end{pmatrix}$$

How to choose the sign of α ? The idea is to let we have less round off error (i.e, to induce less subtractive cancelation). Thus choose the sign as $-$ sgn a_{11}.

QR with Householder matrix-general

$$\begin{bmatrix} I_r & \mathbf{0} \\ \mathbf{0} & H_{r+1} \end{bmatrix} \begin{bmatrix} X_a & X_b \\ \mathbf{0} & X_c \end{bmatrix} = \begin{bmatrix} X_\alpha & X_\beta \\ \mathbf{0} & X_\gamma \end{bmatrix}$$

where H_{r+1} is an $(m-r)\times(m-r)$ elementary reflector. We choose the sign of α_1 opposite to that of the first entry of $\mathbf{a}_1$ for stability(less round off errors).

2.8 Gram-Schmidt Orthogonalization and the QR Decomposition

Classical Gram-Schmidt

Given n linearly independent vectors $\{\mathbf{v}_1, \cdots, \mathbf{v}_n\}$, we consider the span $V = span\{\mathbf{v}_1, \cdots, \mathbf{v}_n\}$ and we would like to find an orthonormal basis for V. The

usual Gram-Schmidt process is to find $\mathbf{u}_1, \cdots, \mathbf{u}_n$ as follows:

$$\begin{array}{rcll} \mathbf{u}_1 &=& \frac{\mathbf{v}_1}{\|\mathbf{v}_1\|} & \\ \mathbf{u}_2' &=& \mathbf{v}_2 - (\mathbf{v}_2, \mathbf{u}_1)\mathbf{u}_1, & \mathbf{u}_2 = \frac{\mathbf{u}_2'}{\|\mathbf{u}_2'\|} \\ \mathbf{u}_3' &=& \mathbf{v}_3 - (\mathbf{v}_3, \mathbf{u}_2)\mathbf{u}_2 - (\mathbf{v}_3, \mathbf{u}_1)\mathbf{u}_1, & \mathbf{u}_3 = \frac{\mathbf{u}_3'}{\|\mathbf{u}_3'\|} \\ &=& \cdots & \\ \mathbf{u}_n' &=& \mathbf{v}_n - \sum_{j=1}^{n-1}(\mathbf{v}_k, \mathbf{u}_j)\mathbf{u}_j, & \mathbf{u}_n = \frac{\mathbf{u}_n'}{\|\mathbf{u}_n'\|}. \end{array} \tag{2.23}$$

Assume A is a $m \times m$ square matrix whose columns are $\{\mathbf{v}_1, \cdots, \mathbf{v}_m\}$. Writing this process in the matrix form, we see $AR = Q$ where columns of Q are $\mathbf{u}_1, \cdots, \mathbf{u}_m$ and columns of R are $\mathbf{r}_1, \mathbf{r}_2, \cdots$, which form a triangular matrix. Thus $A = QR'$ is another QR-decomposition of A.

Modified Gram-Schmidt

Unfortunately, this is numerically bad procedure: round off errors and loss of orthogonality of Q. For better numerical behavior, use the following modified Gram-Schmidt. The idea is to use the orthogonalize all vectors w.r.t orthogonal vector $\mathbf{q}_i$ as soon as they are available.

Given linearly independent vectors $\{\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_n\}$,

$$\begin{array}{rcl} \{\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_n\} & \xrightarrow{\text{normalize}} & \{\mathbf{q}_1, \mathbf{v}_2, \cdots, \mathbf{v}_n\} \\ & \xrightarrow{\text{apply } P_{\perp \mathbf{q}_1}} & \{\mathbf{q}_1, P_{\perp \mathbf{q}_1}\mathbf{v}_2, \cdots, P_{\perp \mathbf{q}_1}\mathbf{v}_n\} \\ & \xrightarrow{\text{normalize}} & \{\mathbf{q}_1, \mathbf{q}_2, \mathbf{v}_3, \cdots, P_{\perp \mathbf{q}_1}\mathbf{v}_n\} \\ & \xrightarrow{\text{apply } P_{\perp \mathbf{q}_2}} & \{\mathbf{q}_1, \mathbf{q}_2, P_{\perp \mathbf{q}_2}P_{\perp \mathbf{q}_1}\mathbf{v}_3, \cdots, P_{\perp \mathbf{q}_2}P_{\perp \mathbf{q}_1}\mathbf{v}_n\} \\ & \xrightarrow{\text{normalize}} & \{\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3, \cdots, P_{\perp \mathbf{q}_2}P_{\perp \mathbf{q}_1}\mathbf{v}_n\} \\ & \vdots & \vdots \end{array}$$

CGS-**Algorithm**: Given $\mathbf{v}_1, \cdots, \mathbf{v}_n \in \mathbb{R}^m$

For $k = 1, 2, \ldots, n$

$s_{ik} = \mathbf{q}_i^T \mathbf{v}_k, (i = 1, 2, \ldots, k-1)$

$\mathbf{v}_k \leftarrow \mathbf{v}_k - \sum_{i=1}^{k-1} s_{ik}\mathbf{q}_i (\mathbf{v}_k \leftarrow \mathbf{v}_k - \sum_{i=1}^{k-1} < \mathbf{q}_i, \mathbf{v}_k > \mathbf{q}_i)$

$r_{kk} = \|\mathbf{z}_k\|, \ \mathbf{q}_k = \frac{\mathbf{v}_k}{r_{kk}}$

$r_{ik} = \frac{s_{ik}}{r_{kk}}, (i = 1, 2, \ldots, k-1)$

Modified-GS Algorithm: Given $\mathbf{v}_1, \cdots, \mathbf{v}_n \in \mathbb{R}^m$

For $k = 1, 2, \ldots, n$

$r_{kk} = \|\mathbf{v}_k\|$

For $i = 1, 2, \ldots, m$

$q_{ik} = \frac{v_{ik}}{r_{kk}}$ ($\mathbf{q}_k = \frac{\mathbf{v}_k}{r_{kk}}$ normalize k-th vector $\mathbf{v}_k$)

End i-loop(Note the whole process ends here)

For $j = k+1, \ldots, n$

$r_{kj} = \sum_{i=1}^{m} q_{ik} v_{ij}$ $(= \mathbf{q}_k^T \cdot \mathbf{v}_j)$

For $i = 1, 2, \ldots, m$

$v_{ij} \leftarrow v_{ij} - v_{ik} r_{kj}$ $(\mathbf{v}_j \leftarrow \mathbf{v}_j - < \mathbf{q}_k, \mathbf{v}_j > \mathbf{q}_k)$

End i-loop

End j-loop

Here $Q = \{q_{ij}\}$ and $R = \{r_{ij}\}$. To save space A is overwritten by Q-orthogonal vectors.

Example 2.8.1. Apply MGS to A find the QR-decomposition, where

$$A = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}.$$

Note that $\mathbf{v}_1 = (1,0,1)^T, \mathbf{v}_2 = (-1,1,1)^T, \mathbf{v}_2 = (0,1,1)^T$.

Modified-GS Algorithm for A

$k = 1$. Normalize first column. $r_{11} = \|\mathbf{v}_1\| = \sqrt{2}$. So $\mathbf{q}_1 = \frac{1}{\sqrt{2}}(1,0,1)^T$.

$j = 2$

$r_{12} = \mathbf{q}_1^T \cdot \mathbf{v}_2 = \frac{1}{\sqrt{2}}(1,0,1) \cdot (-1,1,1) = 0.$

$\mathbf{v}_2 \leftarrow \mathbf{v}_2 - 0 \cdot \mathbf{q}_1$

$j = 3$

$r_{13} = \mathbf{q}_1^T \cdot \mathbf{v}_3 = \frac{1}{\sqrt{2}}(1,0,1) \cdot (0,1,1) = \frac{1}{\sqrt{2}}.$

$\mathbf{v}_3 \leftarrow \mathbf{v}_3 - r_{13}\mathbf{q}_1 = (0,1,1)^T - \frac{1}{\sqrt{2}}\frac{1}{\sqrt{2}}(1,0,1)^T = (-\frac{1}{2}, 1, \frac{1}{2})^T$

$k = 2$. Normalize second column. $r_{22} = \|\mathbf{v}_2\| = \sqrt{3}$. So $\mathbf{q}_2 = \frac{1}{\sqrt{3}}(-1,1,1)^T$.

$j = 3$

$r_{23} = \mathbf{q}_2^T \cdot \mathbf{v}_3 = \frac{1}{\sqrt{3}}(-1,1,1) \cdot (-\frac{1}{2}, 1, \frac{1}{2}) = \frac{2}{\sqrt{3}}.$

$\mathbf{v}_3 \leftarrow \mathbf{v}_3 - r_{23}\mathbf{q}_2 = (-\frac{1}{2}, 1, \frac{1}{2})^T - \frac{2}{\sqrt{3}}\frac{1}{\sqrt{3}}(-1,1,1)^T = \frac{1}{6}(1,2,-1)^T$

$k = 3$. Normalize third column. $r_{33} = \|\mathbf{v}_3\| = \frac{1}{\sqrt{6}}$.

So $\mathbf{q}_3 = \frac{1}{\sqrt{6}}(1,2,-1)^T$. □

At the end of each $k = 1, 2, 3$, we have

$$AR_1 = Q_= \begin{bmatrix} \frac{1}{\sqrt{2}} & -1 & -\frac{1}{2} \\ 0 & 1 & 1 \\ \frac{1}{\sqrt{2}} & 1 & \frac{1}{2} \end{bmatrix}, \; R_1 = \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -\frac{1}{\sqrt{2}} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & -\frac{1}{2} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$AR_1R_2 = Q_2 = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{3}} & -\frac{1}{2} \\ 0 & \frac{1}{\sqrt{3}} & 1 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{3}} & \frac{1}{2} \end{bmatrix}, \; R_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{\sqrt{3}} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -\frac{2}{\sqrt{3}} \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{\sqrt{3}} & -\frac{2}{3} \\ 0 & 0 & 1 \end{bmatrix}$$

$$AR_1R_2R_3 = Q_3 := Q = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{6}} \\ 0 & \frac{1}{\sqrt{3}} & \frac{2}{\sqrt{6}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{6}} \end{bmatrix}, \; R_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \sqrt{6} \end{bmatrix}.$$

Hence

$$A = QR_3^{-1}R_1^{-2}R_1^{-1} = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{6}} \\ 0 & \frac{1}{\sqrt{3}} & \frac{2}{\sqrt{6}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{6}} \end{bmatrix} \cdot \begin{bmatrix} \sqrt{2} & 0 & \frac{1}{\sqrt{2}} \\ 0 & \sqrt{3} & \frac{2}{\sqrt{3}} \\ 0 & 0 & \frac{1}{\sqrt{6}} \end{bmatrix}$$

Here

$$R_3^{-1}R_2^{-1}R_1^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \frac{1}{\sqrt{6}} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \sqrt{3} & \frac{2}{\sqrt{3}} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \sqrt{2} & 0 & -\frac{1}{\sqrt{2}} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \sqrt{2} & 0 & \frac{1}{\sqrt{2}} \\ 0 & \sqrt{3} & \frac{2}{\sqrt{3}} \\ 0 & 0 & \frac{1}{\sqrt{6}} \end{bmatrix}$$

Thus, the entries of $R_3^{-1}R_2^{-1}R_1^{-1}$ coincides with above $R = \{r_{ij}\}$, i.e,

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ 0 & r_{22} & r_{23} \\ 0 & 0 & r_{23} \end{bmatrix} = \begin{bmatrix} \sqrt{2} & 0 & \frac{1}{\sqrt{2}} \\ 0 & \sqrt{3} & \frac{2}{\sqrt{3}} \\ 0 & 0 & \frac{1}{\sqrt{6}} \end{bmatrix}$$

2.9 SVD

Theorem 2.9.1 (Singular value decomposition)**.** *If $A \in \mathbb{R}^{m\times n}$ then there exists orthogonal matrices $U \in \mathbb{R}^{m\times m}$ and $V \in \mathbb{R}^{n\times n}$ such that*

$$U^t AV = \Sigma = \text{diag}(\sigma_1, \sigma_2, \cdots, \sigma_p) \tag{2.24}$$

where Σ is $m \times n$ matrix and $p = \min(m, n)$ and $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p \geq 0$.

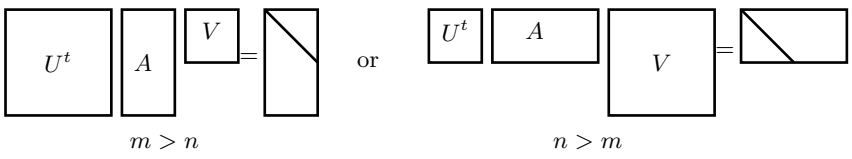

Figure 2.6: Σ of SVD

The σ_i are the singular values of A and vectors $\mathbf{u}_i$ and $\mathbf{v}_i$ are left singular

and right singular vectors. We see

$$\begin{matrix} A\mathbf{v}_i = \sigma_i \mathbf{u}_i \\ A^T \mathbf{u}_i = \sigma_i \mathbf{v}_i \end{matrix} , \quad i = 1, \cdots, p$$

Consider solving an over-determined system $A\mathbf{x} = \mathbf{b}$ by least square method. We need to solve a normal equation $A^T A\mathbf{x} = A^T \mathbf{b}$. Thus

$$\begin{aligned} (U\Sigma V^T)^T U\Sigma V^T \mathbf{x} &= (U\Sigma V^T)^T \mathbf{b} \\ V\Sigma^T U^T U\Sigma V^T \mathbf{x} &= V\Sigma^T U^T \mathbf{b} \\ V\Sigma^T \Sigma V^T \mathbf{x} &= V\Sigma^T U^T \mathbf{b} \\ \Sigma^T \Sigma V^T \mathbf{x} &= \Sigma^T U^T \mathbf{b} \\ V^T \mathbf{x} &= \Sigma^{-1} U^T \mathbf{b} \\ \mathbf{x} &= V\Sigma^{-1} U^T \mathbf{b}. \end{aligned}$$

In general, SVD is expansive.