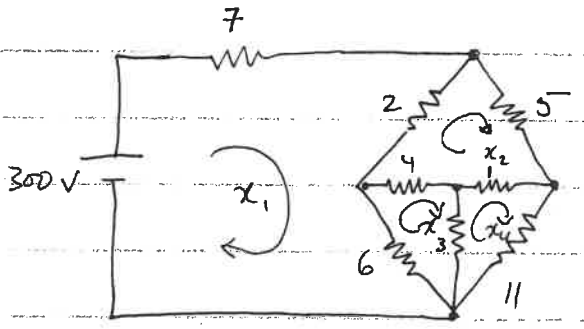
Chapter 6 : Systems of Linear Equations

Ex. Linear system arising from electrical circuits:

x_1, x_2, x_3, x_4 are "loop" currents

$$15x_1 - 2x_2 - 6x_3 = 300$$
$$-2x_1 + 12x_2 - 4x_3 - x_4 = 0$$
$$-6x_1 - 4x_2 + 19x_3 - 9x_4 = 0$$
$$-x_2 - 9x_3 + 21x_4 = 0$$



we can write this system of 4 equations in 4 unknowns in matrix-vector form

$$\begin{bmatrix} 15 & -2 & -6 & 0 \\ -2 & 12 & -4 & -1 \\ -6 & -4 & 19 & -9 \\ 0 & -1 & -9 & 21 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 300 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$Ax = b$$

In this chapter we will study

(i) Various methods for solving $Ax=b$

(ii) special methods for matrices A with special structure

Methods — Direct: Gauss Elimination

Iterative: Jacobi, Gauss-Seidel, SOR, Conjugate Gradient

Special structures & properties: triangular, tridiagonal, banded, positive definite, sparse, ...

Review of some basic facts from matrix Algebra

Defn An $m \times n$ matrix A is a rectangular array of elements

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix}$$ m rows and n columns

$(A)_{ij} = a_{ij}$

Defn If A and B are both $m \times n$ matrices, then the sum of A denoted $A+B$ is the $m \times n$ matrix whose entries are $a_{ij} + b_{ij}$ $i = 1, \ldots, m$, $j = 1, \ldots, n$.

similarly, $(A - B)_{ij} = a_{ij} - b_{ij}$.

Defn If A is an $m \times n$ matrix, and λ a real number, then the scalar multiplication of λ and A, denoted λA is the $m \times n$ matrix whose entries are λa_{ij}, $i = 1, \ldots, m$; $j = 1, \ldots, n$.

Theorem

(i) $A + B = B + A$

(ii) $A + 0 = 0 + A = A$ $\quad 0 = \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & & \vdots \\ 0 & \cdots & 0 \end{bmatrix}$ $\rightarrow$ the $m \times n$ zero matrix

(iii) $A + (B + C) = (A + B) + C$

(iv) $A + (-A) = (-A) + A = 0$

(v) $\lambda(A + B) = \lambda A + \lambda B$

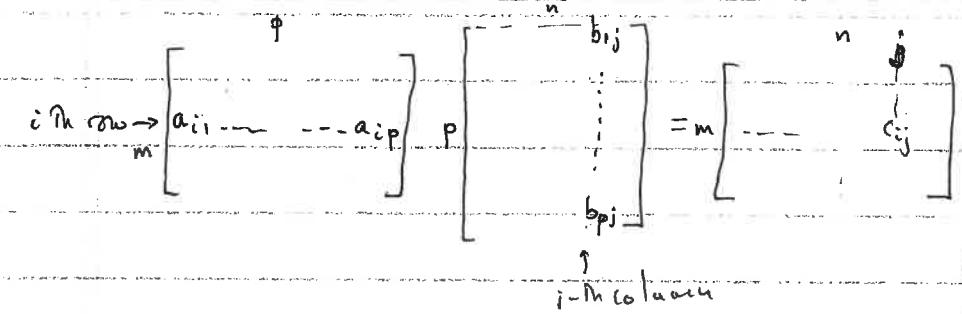
(vi) $(\lambda + \mu) A = \lambda A + \mu A$

(vii) $\lambda(\mu A) = \lambda \mu A$

(viii) $1 \cdot A = A$.

Defn. Let A be an $m \times p$ matrix and let B be a $p \times n$ matrix. The **matrix product** of A and B denoted AB, is an $m \times n$ matrix C whose ij-th element is given by

$$c_{ij} = \sum_{k=1}^{p} a_{ik} b_{kj}$$



(a) $A(BC) = (AB)C$

(b) $A(B+D) = AB + AD$

(c) $\lambda(AB) = (\lambda A)B = A(\lambda B)$

Note (i) $AB \neq BA$ in general — order is important. BA may not even be defined.

(ii) 2nd dim of A = 1st dim of B

Defn. An $m \times n$ matrix is called **square** if $m = n$.

Defn. A square matrix A is called **diagonal** if $a_{ij} = 0$ if $i \neq j$.

Defn. A square matrix A is called **upper triangular** if $a_{ij} = 0$ if $i > j$

Defn. " " " " " **lower** " if $a_{ij} = 0$ if $i < j$.

Defn. A square matrix A is **symmetric** if $a_{ij} = a_{ji}$.

Determinants

Let A be a square matrix. Its determinant (denoted $\det(A)$ or $|A|$) is given by

$$\det(A) = \sum (-1)^{\sigma} a_{1 i_1} a_{2 i_2} \cdots a_{n i_n}$$

where the sum runs over all possible permutations

$(i_1, i_2, \ldots, i_n)$ of $(1, 2, \ldots, n)$ ($n!$ in total) and σ is the number of basic permutations required to bring into the natural ordering $(1, 2, \ldots, n)$

Ex. $A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$, $\det A = (-1)^{\sigma_1} a_{11} a_{22} + (-1)^{\sigma_2} a_{12} a_{21} = a_{11} a_{12} - a_{12} a_{21}$

$(1,2) \to (1,2)$ $\sigma_1 = 0$; $(2,1) \to (1,2)$ $\sigma_2 = -1$

Ex. $A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$

$\det(A) = (-1)^0 a_{11} a_{22} a_{33} + (-1)^1 a_{11} a_{23} a_{32} + (-1)^1 a_{12} a_{21} a_{33}$

$(-1)^2 a_{12} a_{23} a_{31} + (-1)^2 a_{13} a_{21} a_{32} + (-1)^3 a_{13} a_{22} a_{31}$

$= a_{11} a_{22} a_{33} - a_{11} a_{23} a_{32} - a_{12} a_{21} a_{33} + a_{12} a_{23} a_{31} + a_{13} a_{21} a_{32} - a_{13} a_{22} a_{31}$

$(1,2,3) \to \sigma = 0$

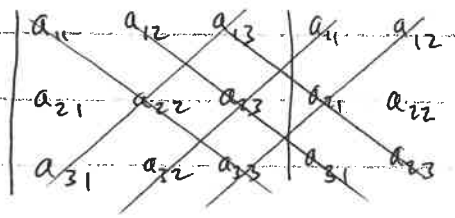
$(1,3,2) \to \sigma = 1$

$(2,1,3) \to \sigma = 1$

$(2,3,1) \to \sigma = 2$

$(3,1,2) \to \sigma = 2$

$(3,2,1) \to \sigma = 3$

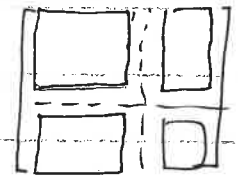


$a_{11} a_{22} a_{33} + a_{12} a_{23} a_{31} + a_{13} a_{21} a_{32}$ $\to$

$= a_{31} a_{22} a_{13} - a_{32} a_{23} a_{11} - a_{33} a_{21} a_{12}$

It is also possible to compute determinants using cofactor expansions.

(1) If A is an $n \times n$ matrix, the minor M_{ij} is the determinant of the $(n-1) \times (n-1)$ submatrix of A obtained by deleting the $i=$th row and $j=$th column of A.

$M_{ij} = \det$ 

(2) The cofactor A_{ij} associated with M_{ij} is $A_{ij} = (-1)^{i+j} M_{ij}$

(3) $\det A = \sum_{i=1}^{n} a_{ij} A_{ij} = \sum_{i=1}^{n} (-1)^{i+j} a_{ij} M_{ij}$, for any $j = 1, 2, \ldots, n$, fixed

or

$\det A = \sum_{j=1}^{n} a_{ij} A_{ij} = \sum_{j=1}^{n} (-1)^{i+j} M_{ij}$, for any $i = 1, 2, \ldots, n$ fixed

Facts about determinants

(a) If any row or column of A has only zeros, then $\det A = 0$.

(b) If any two rows or columns of A are the same, then $\det A = 0$.

(c) If $\tilde{A}$ is obtained from A by permuting two rows (or two columns) then $\det \tilde{A} = -\det A$.

(d) If $\tilde{A}$ is obtained from A by multiplying a given row (or column) by λ then $\det \tilde{A} = \lambda \det A$

$\Rightarrow \det(\lambda A) = \lambda^n \det(A)$

(e) If $\tilde{A}$ is obtained from A by adding to row j a multiple λ of row i, then $\det \tilde{A} = \det A$.

(f) If A is triangular then $\det A =$ product of diagonal elements

(g) If A and B are both $n \times n$ matrices, then $\det(AB) = \det A \cdot \det B$ (note $\det(A+B) \neq \det A + \det B$ in general).

Remark

In view of (f) and (g), the most efficient way of calculating a determinant is to bring A into triangular form.

Defn. An $n \times n$ matrix A is said to be nonsingular or invertible if an $n \times n$ matrix, denoted A^{-1}, exists with $AA^{-1} = A^{-1}A = I$. A^{-1} is called the inverse of A.

Theorem For any nonsingular matrix A

(a) A^{-1} is unique

(b) A^{-1} is invertible and $(A^{-1})^{-1} = A$

(c) If A & B are two $n \times n$ invertible matrices, then so is AB and $(AB)^{-1} = B^{-1}A^{-1}$

Defn. The transpose A^T of an $n \times n$ matrix A is the $n \times n$ matrix such that $(A^T)_{ij} = (A)_{ji}$.

Defn. A square matrix is symmetric if $A^T = A$, i.e. $a_{ij} = a_{ji}$

Theorem.

(a) $(A^T)^T = A$

(b) $(A+B)^T = A^T + B^T$

(c) $(AB)^T = B^T A^T$

(d) If A^{-1} exists then $(A^{-1})^T = (A^T)^{-1}$.

Theorem If A is square then A^{-1} exists $\iff \det A \neq 0$

Theorem If A^{-1} exists, then $\det(A^{-1}) = 1/\det(A)$.

Theorem Consider the system $Ax = b$, where A is $n \times n$. Then for any $b \in \mathbb{R}^n$ the system has a unique solution if and only if A^{-1} exists.

Conditioning of linear systems

Before embarking on a study of various algorithms for solving a linear system $Ax=b$, it is important to take into account the following considerations

1) The entries of A and b are often produced by calculations or even experiments, so they involve some uncertainties or as some statisticians would refer to such: "noise".

2) Even when we are "certain" of the entries of A and b, when we "enter" them into the computer, they will be represented by "nearby" numbers due to the finite precision arithmetic.

So, almost invariably, we are solving a "nearby" system or perturbed

$$(A+\delta A)(x+\delta x) = b+\delta b \quad \text{vs.} \quad Ax=b.$$

> The question is then: Can δx be large even if δA and δb are small?

Finding answers to this question is of extreme importance. To take steps in the right direction we need tools to measure vectors and matrices, in other words, norms we assume the reader is familiar with these concepts.

Theorem Consider the system $Ax=b$ and the "perturbed" system $(A+\delta A)(x+\delta x) = b+\delta b$. Assume that A is ~~invertible and $\|A^{-1}\delta A\| < 1$~~. Then $A+\delta A$ is invertible and

$$\text{(*)} \qquad \frac{\|\delta x\|}{\|x\|} \leq \frac{1}{1-\|A^{-1}\delta A\|} \|A\|\,\|A^{-1}\| \left[\frac{\|\delta A\|}{\|A\|} + \frac{\|\delta b\|}{\|b\|}\right].$$

proof. We shall use the following result whose proof is given afterwards

Lemma (Neumann lemma) Suppose the matrix E satisfies $\|E\| < 1$ for some natural matrix norm $\|\cdot\|$. Then $I-E$ and $I+E$ are invertible and

$$\|(I \pm E)^{-1}\| \leq \frac{1}{1-\|E\|}.$$

proof. Suppose $I-E$ is not invertible. There, there must exist a non zero vector x such that $(I-E)x=0$. This implies

$$x = Ex \Rightarrow \|x\| = \|Ex\| \leq \|E\|\,\|x\| < \|x\|$$

which is a contradiction. Now consider the identity

$$(I-E)(I+E+\cdots+E^n) = I-E^{n+1}$$

or

$$I+E+\cdots+E^n = (I-E)^{-1} - (I-E)^{-1}E^{n+1}.$$

Since $\|E\|<1$, $\|E^{n+1}\| \leq \|E\|^{n+1}$, converges to zero, hence the right side converges to $(I-E)^{-1}$ implying that the left side converges to $(I-E)^{-1}$.

In fact we have shown that the infinite matrix series $I+E+\cdots+E^n+\cdots$ is convergent and is the inverse of $I-E$. Furthermore

$$\|(I-E)^{-1}\| = \|I+E+\cdots+E^n+\cdots\|$$

$$\leq \|I\| + \|E\| + \cdots + \|E^n\| + \cdots$$

$$\leq \frac{1}{1-\|E\|}.$$

The case of $I+E$ can be treated in a similar way. Except that

$$(I+E)^{-1} = I-E+E^2-E^3+\cdots \qquad \blacksquare$$

Corollary (Banach perturbation lemma) Let A be invertible. If B satisfies $\|A^{-1}B\| < 1$, then $A \pm B$ are invertible and

$$\|(A \pm B)^{-1}\| \leq \frac{\|A^{-1}\|}{1-\|A^{-1}B\|}.$$

proof write $A \pm B = A(I \pm A^{-1}B)$ and use lemma with $E = \mp B$. $\blacksquare$

proof of the theorem. From $(A+\delta A)(x+\delta x)=b+\delta b$, we have

$$Ax+(\delta A)x+A\,\delta x+(\delta A)\delta x=b+\delta b.$$

Subtracting Ax from the left side and b from the right side we get

$$(A+\delta A)\delta x=\delta b-(\delta A)x.$$

By previous corollary $A+\delta A$ is invertible, hence

$$\delta x=(A+\delta A)^{-1}[\delta b-(\delta A)x].$$

Taking norms and using bound on $\|(A+\delta A)^{-1}\|$

$$\|\delta x\|\le\frac{\|A^{-1}\|}{1-\|A^{-1}\delta A\|}\left[\|\delta b\|+\|\delta A\|\,\|x\|\right].$$

$$=\frac{\|A\|\,\|A^{-1}\|}{1-\|A^{-1}\delta A\|}\left[\frac{\|\delta b\|}{\|A\|}+\frac{\|\delta A\|}{\|A\|}\|x\|\right].$$

Divide both sides by $\|x\|$:

$$\frac{\|\delta x\|}{\|x\|}\le\frac{\|A\|\,\|A^{-1}\|}{1-\|A^{-1}\delta A\|}\left[\frac{\|\delta b\|}{\|A\|\,\|x\|}+\frac{\|\delta A\|}{\|A\|}\right].$$

From $b=Ax$ we have $\|b\|=\|Ax\|\le\|A\|\,\|x\|$. This allows us to replace $\|A\|\,\|x\|$ in the denominator by $\|b\|$ and gives the desired bound on $\frac{\|\delta x\|}{\|x\|}$. ∎

In order to interpret this result, let us assume that $\|A^{-1}\delta A\|<\frac{1}{2}$, which is reasonable. Thus we have

$$\frac{\|\delta x\|}{\|x\|}\le 2\,\|A\|\,\|A^{-1}\|\left[\frac{\|\delta b\|}{\|b\|}+\frac{\|\delta A\|}{\|A\|}\right].$$

The quantity $\|A\|\,\|A^{-1}\|$ plays a very important role in numerical linear algebra. It is called the condition number of A.

More generally, given a square matrix A and a natural matrix norm $\|\cdot\|$, we define $\mathrm{cond}_{\|\cdot\|}(A)$ by

$$\mathrm{cond}_{\|\cdot\|}(A)=\begin{cases}\infty & \text{if } A \text{ is singular}\\ \|A\|\,\|A^{-1}\| & \text{if } A \text{ is invertible.}\end{cases}$$

Indeed the relative error $\frac{\|\delta x\|}{\|x\|}$ may be as large as the relative errors $\frac{\|\delta b\|}{\|b\|}$ and $\frac{\|\delta A\|}{\|A\|}$ magnified by the factor $\text{cond}(A)$. Of course, since we are dealing with an inequality, the left side, i.e. $\frac{\|\delta x\|}{\|x\|}$ may happen to be much smaller than the right side. On the other hand if $\text{cond}(A)$ is "large" even small perturbations in b and/or A may cause large errors in the solution x.

In any norm, $1 \le \text{cond}(A) \le \infty$ and for large values of $\text{cond}(A)$, the matrix A is said to be ill-conditioned. Here "large" must be interpreted in somewhat subjective terms. For example if $\text{cond}(A) = 100$, the estimate (*) shows that relative changes of 10^{-8} in the right side may cause relative errors in the solution of 2×10^{-6} which may or may not be considered to be a large degradation in accuracy.

Ex. let $A = \begin{bmatrix} 1 & 1 \\ 1 & 1+\epsilon \end{bmatrix}$, $0 < |\epsilon| < 1$

$$\|A\|_\infty = 2 + \epsilon, \quad A^{-1} = \frac{1}{\epsilon}\begin{bmatrix} 1+\epsilon & -1 \\ -1 & 1 \end{bmatrix} \Rightarrow \|A^{-1}\|_\infty = \frac{2+\epsilon}{|\epsilon|}$$

$$\Rightarrow \text{cond}_\infty(A) = \frac{(2+\epsilon)^2}{|\epsilon|} \approx \frac{4}{|\epsilon|}.$$

Hence for $|\epsilon|$ small, A can be considered to be ill-conditioned with severity depending on how small ϵ is.

Two families of matrices known to be notoriously ill-conditioned are the Hilbert and Vandermonde matrices. For $n \ge 1$, the Hilbert matrix H_n is given by

$$H_n = \begin{bmatrix} 1 & \frac{1}{2} & \cdots & \frac{1}{n} \\ \frac{1}{2} & & & \\ \vdots & & & \\ \frac{1}{n} & \cdots & & \frac{1}{2n-1} \end{bmatrix}.$$

It can be shown that (G. Szegö) (see Gautschi)

$$\text{Cond}_2(H_n) \sim \frac{(\sqrt{2}+1)^{4n+4}}{2^{15/4}\sqrt{\pi n}} \Rightarrow$$

n	10	20	40
$\text{Cond}_2(H_n)$	1.6×10^{13}	2.45×10^{28}	7.65×10^{58}

Hilbert matrices arise in least-squares approximations of functions.

Vandermonde Matrices are of the form

$$V_n = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ t_1 & t_2 & \cdots & t_n \\ \vdots & \vdots & & \vdots \\ t_1^{n-1} & t_2^{n-1} & \cdots & t_n^{n-1} \end{bmatrix}, \quad t_1, \ldots, t_n \text{ distinct points.}$$

$$t_k = 1 - \frac{2(k-1)}{n-1}, \quad k = 1, \ldots, n.$$

and arise in interpolation (polynomials). It can be shown that

$$\text{cond}_\infty(V_n) \sim \frac{1}{\pi} e^{-\frac{\pi}{4}} e^{n\left(\frac{\pi}{4} + \frac{1}{2}\ln 2\right)} \Rightarrow$$

n	10	20	40
$\text{cond}_\infty(V_n)$	1.36×10^{4}	1.05×10^{9}	6.93×10^{18}

If $t_k = \frac{1}{k}$, $k = 1, \ldots, n$; then $\text{cond}_\infty(V_n) > n^{n+1}$

Solving linear systems with these matrices is basically a hopeless task. Actual computations show that the relative errors in x obey the theoretical bounds.

Fortunately, the same problems that led to the Hilbert and Vandermonde matrices can be cast and solved in ways that are not ill-conditioned. For example, Hilbert matrices arise upon using $\{1, x, \ldots, x^n\}$ as basis functions. Using a family of orthogonal basis functions e.g. Legendre polynomials leads to linear systems that are well or even perfectly conditioned.

§6.1 Naive Gaussian Elimination

Consider, as an example, the system

$$6x_1 - 2x_2 + 2x_3 + 4x_4 = 16$$
$$12x_1 - 8x_2 + 6x_3 + 10x_4 = 26$$
$$3x_1 - 13x_2 + 9x_3 + 3x_4 = -19$$
$$-6x_1 + 4x_2 + x_3 - 18x_4 = -34$$

We will do away with the variables, then attach the right side as an extra column $\rightarrow$ augmented system

$$\left[\begin{array}{cccc|c} \textcircled{6} & -2 & 2 & 4 & 16 \\ 12 & -8 & 6 & 10 & 26 \\ 3 & -13 & 9 & 3 & -19 \\ -6 & 4 & 1 & -18 & -34 \end{array}\right]$$

In naive Gaussian Elimination, we use nonzero elements along the diagonal, called pivots to introduce zeros below the pivot.

6 is pivot multiply 1st row by -2 and add to 2nd

" " " " $-\frac{1}{2}$ " " 3rd

" " " " 1 " " 4th

$$\Rightarrow \left[\begin{array}{cccc|c} 6 & -2 & 2 & 4 & 16 \\ 0 & \textcircled{-4} & 2 & 2 & -6 \\ 0 & -12 & 8 & 1 & -27 \\ 0 & 2 & 3 & -14 & -18 \end{array}\right]$$

Next, we use -4 as pivot to eliminate elements in position (3,2) and (4,2)

-4 is pivot — mult 2nd row by -3 and add to 3rd

" 2nd row by $\frac{1}{2}$ " " " " 4th

=>

$$\left[\begin{array}{cccc|c} 6 & -2 & 2 & 4 & 16 \\ 0 & -4 & 2 & 2 & -6 \\ 0 & 0 & \textcircled{2} & -5 & -9 \\ 0 & 0 & 4 & -13 & -21 \end{array}\right]$$

Finally, using 2, as pivot,

we multiply row 3 by -2 and add to 4th

=>

$$\left[\begin{array}{cccc|c} 6 & -2 & 2 & 4 & 16 \\ 0 & -4 & 2 & 2 & -6 \\ 0 & 0 & 2 & -5 & -9 \\ 0 & 0 & 0 & -3 & -3 \end{array}\right]$$

The purpose of G.E. is to bring system into <u>upper triangular form</u>. Once a system is in this form, the unknowns can be solved for in <u>backward</u> order <u>back substitution</u>

Indeed, 4th row => $-3x_4 = -3$ => $\boxed{x_4 = 1}$

3rd row => $2x_3 - 5x_4 = -9$ => $x_3 = \frac{5x_4 - 9}{2} = \frac{5-9}{2} = -2$

2nd row => $-4x_2 + 2x_3 + 2x_4 = -6$ => $x_2 = \frac{-6 - 2x_3 - 2x_4}{-4} = 1$

1st " => $6x_1 - 2x_2 + 2x_3 + 4x_4 = 16$ => $x_1 = (16 - 2x_2 - 2x_3 - 4x_4)/6 = 3$

Algorithm we will view Gauss Elimination as a sequence of operations applied to $[A \mid b]$

$$[A^{(1)} \mid b^{(1)}] = [A \mid b] \qquad \text{Initialization}$$

$$[A^{(2)} \mid b^{(2)}] = \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} & b_1^{(1)} \\ 0 & a_{22}^{(2)} & & a_{2n}^{(2)} & b_2^{(2)} \\ \vdots & & & & \vdots \\ 0 & a_{n2}^{(2)} & \cdots & a_{nn}^{(2)} & b_n^{(2)} \end{bmatrix}$$

After step k,

$$[A^{(k)} \mid b^{(k)}] = \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} & b_1^{(1)} \\ 0 & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} & b_2^{(2)} \\ & & \ddots & & \vdots \\ & & a_{kk}^{(k)} & a_{kn}^{(k)} & b_k^{(k)} \\ & 0 & \vdots & \vdots & \vdots \\ 0 & & a_{nk}^{(k)} & a_{nn}^{(k)} & b_n^{(k)} \end{bmatrix}$$

The algorithm consists in describing how to get $[A^{(k+1)} \mid b^{(k+1)}]$ from $[A^{(k)}, b^{(k)}]$

(i) Elements up to an including row k are not changed.

(ii) Using $a_{kk}^{(k)}$ as pivot (assume $a_{kk}^{(k)} \neq 0$), we introduce zeros in the locations $(k+1, k) \cdots (n, k)$. To introduce a zero at the (i,k) location, $i = k+1, \cdots, n$,

multiply row k by $m_{ik} = \dfrac{-a_{ik}^{(k)}}{a_{kk}^{(k)}}$ and add to row i.

or equivalently

multiply row k by $\tilde{m}_{ik} = \dfrac{a_{ik}^{(k)}}{a_{kk}^{(k)}}$ and subtract from row i.

The numbers m_{ik} (or $\tilde{m}_{ik}$) are called the multipliers

(iii) The remaining elements are changed accordingly

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} a_{kj}^{(k)}, \qquad \begin{matrix} i = k+1, \dots, n \\ j = k+1, \dots, n \end{matrix}$$

$$\Rightarrow \quad \left[A^{(k+1)} \vdots b^{(k+1)}\right] = \left[\begin{array}{cccc|c} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} & b_1^{(1)} \\ & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} & \\ & & \ddots & \vdots & \\ & & a_{kk}^{(k)} \; a_{k,k+1}^{(k)} \cdots & a_{k,n}^{(k)} & b_k^{(k)} \\ & & 0 \mid a_{k+1,k+1}^{(k+1)} \cdots & a_{k+1,n}^{(k+1)} & b_{k+1}^{(k+1)} \\ & & \vdots \qquad \vdots & \vdots & \vdots \\ & & 0 \mid a_{n,k+1}^{(k+1)} \cdots & a_{n,n}^{(k+1)} & b_n^{(k+1)} \end{array}\right]$$

After $n-1$ steps, $k=n$ and

$$\left[A^{(n)} \vdots b^{(n)}\right] = \left[\begin{array}{cccc|c} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} & b_1^{(1)} \\ & a_{22}^{(2)} & \cdots & \vdots & b_2^{(2)} \\ & & \ddots & \vdots & \vdots \\ & & 0 & a_{nn}^{(n)} & b_n^{(n)} \end{array}\right]$$

so that $A^{(n)}$ is in <u>upper triangular form</u>. This is the purpose of Gauss Elimination.

Algorithm: Naïve Gaussian Elimination

```
For k = 1 : n-1
    For i = k+1 : n
        mult = aik / akk
        for j = k+1 : n
            aij = aij - mult * akj
        }
        bi = bi - mult * bk
    }
}
```

At the end of the algorithm, the augmented system is in upper triangular form. The solution x can be calculated by Back substitution.

write

$$U \equiv A^{(n)} = \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} \\ & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} \\ & & \ddots & \vdots \\ 0 & & & a_{nn}^{(n)} \end{bmatrix}; \quad b \equiv b^{(n)} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

Back Substitution

```
x_n = b_n / u_nn
For i = n-1, ..., 1
    sum = 0
    For j = i+1 : n
        sum = sum + u_ij * x_j
    }
    x_i = b_i / sum
}
```

Work Estimates

Elimination: For each i: $(n-k)\left(\underbrace{1}_{\text{for mult}} + \underbrace{2}_{\text{for } b_i} + \underbrace{2(n-k)}_{\text{innermost loop}}\right)$

$$= 3(n-k) + 2(n-k)^2$$

$$\text{work} = \sum_{k=1}^{n-1} \left[3(n-k) + 2(n-k)^2\right] = 3\left(1+2+\cdots+n-1\right) + 2\left(1^2+2^2+\cdots+(n-1)^2\right)$$

$$= 3\frac{(n-1)(n)}{2} + 2\cdot\frac{1}{6}(n-1)(n)(2n-1)$$

$$= \frac{2}{3}n^3 + \frac{n^2}{2} - \frac{7}{6}n = \boxed{\frac{2}{3}n^3 + O(n^2)}$$

Back Substitution

For given i $\longrightarrow$ $(n-i)$ mults + $(n-i-1)$ adds + 1 division $= 2(n-i)$

$$\text{work} = \sum_{i=1}^{n-1} 2(n-i) = 2(1+2+\cdots+n-1) = \frac{2(n-1)n}{2} = \boxed{n^2 - n}$$

At this point we consider the important issue of the successful completion of naive Gaussian Elimination. This turns out to be a subtle issue and as we shall see later the answer depends on whether the calculations are assumed to be done in exact arithmetic or not.

The example of $A = \begin{bmatrix} 0 & 1 \\ 5 & 2 \end{bmatrix}$ shows that the algorithm fails, in fact at the first step!

Furthermore, it is easy to see that the algorithm will successfully complete if and only if all the pivots $a_{kk}^{(k)}$, $k = 1, \ldots, n-1$ are nonzero. We shall next provide a condition on A which guarantees that this will be the case.

Defn. Let A be a square matrix. For k, $k = 1, \ldots, n$ the kth leading principal submatrix of A is the $k \times k$ submatrix of A formed by the intersection of the first k rows and the first k columns of A. The leading principal minors of A are the determinants of the n leading principal submatrices of A.

We shall also need the following property of matrix multiplication in block form. Let A, B and C be three $n \times n$ matrices. We consider the multiplication $C = AB$ in block or partitioned form

$$C = A B$$

$$\left[\begin{array}{c|c} C_{11} & C_{12} \\ \hline C_{21} & C_{22} \end{array}\right] = \left[\begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array}\right] \left[\begin{array}{c|c} B_{11} & B_{12} \\ \hline B_{21} & B_{22} \end{array}\right]$$

It is easy to see that

$$C_{11} = A_{11} B_{11} + A_{12} B_{21}$$
$$C_{12} = A_{11} B_{12} + A_{12} B_{22}$$
$$C_{21} = A_{21} B_{11} + A_{22} B_{21}$$
$$C_{22} = A_{21} B_{12} + A_{22} B_{22}$$

Theorem Let the square matrix A be such that all the leading principal minors are nonzero. Then, naive Gaussian Elimination will terminate successfully.

proof Clearly $a_{11} = a_{11}^{(1)} \neq 0$ since it is a minor. Hence we can use a_{11} to introduce zeros in the first column. Let us then proceed by induction and assume that the pivots $a_{11}^{(1)}, \ldots, a_{k-1,k-1}^{(k-1)}$ encountered have been nonzero leading to

$$A^{(k)} = \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & \cdots & a_{1n}^{(1)} \\ & a_{22}^{(2)} & \cdots & \cdots & a_{2n}^{(2)} \\ & & \ddots & & \vdots \\ & 0 & & a_{kk}^{(k)} & \cdots a_{kn}^{(k)} \\ & & & \vdots & \\ & & & a_{nk}^{(k)} & \cdots a_{nn}^{(k)} \end{bmatrix}.$$

To complete the induction argument, we need to show that $a_{kk}^{(k)} \neq 0$. Now consider the k-th leading principal submatrix of $A^{(k)}$, namely

$$A_{11}^{(k)} = \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1k}^{(1)} \\ & \ddots & & \vdots \\ 0 & & & \\ & & & a_{kk}^{(k)} \end{bmatrix}, \quad \text{using block notation.}$$

This matrix was obtained from the ~~kth~~ leading principal submatrix of A by applying row elimination operations to it. Now such operations do not change determinants. Hence the kth minor of $A^{(k)}$ is equal to the k-th minor of A, hence nonzero. Note that $A_{11}^{(k)}$ is upper triangular, hence

$$0 \neq \det(A_{11}^{(k)}) = a_{11}^{(1)} a_{22}^{(2)} \cdots a_{kk}^{(k)},$$

Thus $a_{kk}^{(k)}$ must be nonzero. ∎

Remark The proof above revealed that all of the leading principal minors of the matrix A are preserved under naive Gaussian elimination.

Remark

The converse of this is "almost" true. Suppose naive Gaussian elimination has terminated successfully; what can we say about the minors of A? Actually if we revisit the proof above, we see that we did not need the last minor, i.e. $\det(A)$, to be nonzero for successful termination. So if the algorithm terminates successfully, we can assert that all the leading principal minors, with the possible exception of $\det(A)$, must be nonzero.

In reality, we cannot ignore the effect of roundoff errors. Indeed, the following example is quite instructive. Let

$$A = \begin{bmatrix} 2 & 1 & 3 \\ 1 & .50001 & -2 \\ 4 & 5 & 3 \end{bmatrix}.$$

The leading principal minors are $2, .00001, 21$, all nonzero. However, if we perform naive Gaussian Elimination using 4 decimal digit arithmetic with rounding, we see that

$$fl(A^{(2)}) = \begin{bmatrix} 2 & 1 & 3 \\ 0 & 0 & -3.5 \\ 0 & 3 & -3 \end{bmatrix},$$

which shows that the algorithm cannot progress further. Of course $fl(A^{(2)})$ is still nonsingular and we can save the situation by interchanging rows 2 and 3.

$$\Rightarrow \begin{bmatrix} 2 & 1 & 3 \\ 0 & 3 & -3 \\ 0 & 0 & -3.5 \end{bmatrix}$$

which is already in upper triangular form. For larger systems, further elimination steps may be needed to carry out the process of triangularization.

Matrix Factorization.

> Matrix factorization consists in writing a matrix A as the product of two or more matrices with simpler structure and/or special properties

As an example of this general principle, let us assume that a square matrix A can be expressed as

$$A = LU$$

where L is lower triangular matrix with unit diagonal elements and U is upper triangular with nonzero diagonal elements. This is called the LU factorization of A.

Let us show some application of this factorization

1) Calculation of the determinant

clearly

$$\det(A) = \det(LU) = \det(L)\det(U)$$

$$= \underbrace{l_{11} \cdot \ldots \cdot l_{nn}}_{1} \cdot u_{11} \cdots u_{nn} = \prod_{i=1}^{n} u_{ii}.$$

This requires $n-1$ multiplications, vs. $O(n!)$ ops. for cofactor expansion.

2) Solution of the system $Ax = b$

If $u_{ii} \neq 0$, $i = 1, \ldots, n$ then A is nonsingular and the system has a unique solution x which can be calculated as follows:

Since

$$Ax = LUx = b$$

we let $\boxed{Y = Ux}$ and then $\boxed{LY = b}$

we have

$$\begin{bmatrix} 1 & & & \\ \ell_{21} & 1 & & \\ \ell_{31} & \ell_{32} & 1 & \\ \vdots & \vdots & & \ddots \\ \ell_{n1} & \ell_{n2} & \cdots & \ell_{n,n-1} \; 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \Leftrightarrow \boxed{y_i = b_i - \sum_{j=1}^{i-1} \ell_{ij}\, y_j}$$

Note: sum is empty for $i = 1$

Algorithm: Forward substitution

```
y(1) = b(1)
for i = 2, ..., n
    sum = 0,
    for j = 1, .., i-1
        sum = sum + l(i,j) * y(j)
    }
    y(i) = b(i) - sum
}
```

work: $n(n-1)$ ops.

Once y has been calculated, we can calculate the solution x from $Ux = y$ using back substitution. This is exactly what was done in naive Gaussian elimination

```
X(n) = y(n) / U(n,n)
for i = n-1, .., 1
    sum = 0
    for j = i+1, .., n
        sum = sum + U(i,j) * X(j)
    }
    X(i) = (Y(i) - sum) / U(i,i)
}
```

work = n^2 ops

Hence, if we are given the L and U factors of A, then we can obtain the solution x using forward and back substitution at a cost of $\sim 2n^2$ ops.

So what is new here? One could argue that we could have calculated x using naive Gaussian Elimination. Indeed, it will turn out that the process of finding the factors L and U and forward and back substitution is equivalent to naive Gaussian Elimination. In fact, applying naive Gaussian Elimination to A will yield L and U.

Before showing how to accomplish this, let us point to an advantage of LU factorization over naive Gaussian Elimination.

Suppose we wish to solve m linear systems with the same coefficient matrix A but with m different right hand side vectors $b^{(1)}, \ldots, b^{(m)}$. Since naive Gaussian Elimination is applied to the <u>augmented</u> system $[A; b]$, the elimination process must involve A for each and every right hand side vector $b^{(i)}$, resulting in a cost of $m\left(\frac{2}{3}n^3 + O(n^2)\right) = \frac{2}{3}mn^3 + O(mn^2)$ ⊛

On the other hand, we will show that the cost of computing L and U is $\frac{2}{3}n^3 + O(n^2)$. Once these have been computed, we can solve for the solution $x^{(i)}$ corresponding to $b^{(i)}$ at a cost of $2n^2$ by forward and backward substitution resulting in a total cost of $\frac{2}{3}n^3 + m(n^2)$ ops.

We shall now apply the $A = LU$ factorization to calculating the inverse A^{-1}. It is not often that A^{-1} is needed. But if it is, we calculate it as follows: we have

$$A A^{-1} = I$$

which means that $A \cdot (j\text{th column of } A^{-1}) = e_j = (0, \ldots, \underset{\substack{\uparrow \\ j}}{1}, 0 \ldots 0)^T$

$j = 1, \ldots, n$

Hence, we factor A into LU at a cost of $\frac{2}{3}n^3 + O(n^2)$ by naive Gaussian elimination and then solve for the

columns of A^{-1} by Forward and back substitutions. The total cost for this approach is ($m=n$ here)

$$\underbrace{\frac{2}{3}n^3 + O(n^2)}_{LU \text{ factorization}} + \underbrace{(2n^2)}_{n \text{ right sides}} = \frac{8}{3}n^3 + O(n^2).$$

If we had done this using naive Gaussian elimination to n augmented systems $[A \mid e_j]_{j=1}^{n}$, the cost would have been, as seen in (*), $\frac{2}{3}n^4$.

In terms of algorithms, naive Gaussian elimination and $A=LU$ factorization are one and the same. Basically

(i) U is the upper triangular part obtained by applying elimination to A

(ii) The strictly lower triangular part of L is formed by the multipliers generated during the elimination process in the exact order of appearance thereof, i.e.

$$l_{ik} = m_{ik} = a_{ik}^{(k)} / a_{kk}^{(k)} \quad 1 \le k \le n-1, \quad k+1 \le i \le n$$

Ex.

$$A^{(1)} = A = \begin{bmatrix} \textcircled{2} & 4 & -2 \\ 4 & 9 & -3 \\ -2 & -3 & 7 \end{bmatrix}, \quad \begin{matrix} m_{21}=2 \\ m_{31}=-1 \end{matrix} \Rightarrow A^{(2)} = \begin{bmatrix} 2 & 4 & -2 \\ 0 & \textcircled{1} & 1 \\ 0 & 1 & 5 \end{bmatrix}$$

$$\Downarrow \; m_{32}=1$$

$$A^{(3)} = \begin{bmatrix} 2 & 4 & -2 \\ 0 & 1 & 1 \\ 0 & 0 & 4 \end{bmatrix}$$

As claimed

$$L = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & 1 & 1 \end{bmatrix}$$

$$U = A^{(3)} = \begin{bmatrix} 2 & 4 & -2 \\ 0 & 1 & 1 \\ 0 & 0 & 4 \end{bmatrix}.$$

Easy to check that LU is indeed equal to A.

Elementary Matrices

Algorithms such as naïve Gaussian Elimination consist of a sequence of operations applied to the rows of a matrix.

We identify three types of row operations which we call elementary. These are

(I) Interchange rows i and j, including the case $i=j$.

(II) Multiply row i by a scalar α. We require α to be non zero, otherwise there is loss of information.

(III) Multiply row i by a scalar μ and add to row j, i.e. row j is replaced by row$j$ $+\mu$ row$i$.

$$a_{jk} \leftarrow a_{jk} + \mu a_{ik}, \quad k=1,\dots,n.$$

With each one of these elementary operations we will associate an Elementary matrix. These are defined by applying the operation to the identity matrix

Type (I)

$$P_{ij} = \begin{bmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 0 & \cdots & 1 & \leftarrow i \\ & & \vdots & \ddots & \vdots & \\ & & 1 & \cdots & 0 & \leftarrow j \\ & & & & & \ddots \\ & & & & & & 1 \end{bmatrix}$$

Type (II)

$$M_{i,\alpha} = \begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & \alpha & & \leftarrow i \\ & & & \ddots & \\ & & & & 1 \end{bmatrix}$$

Type III

$$E^{\mu}_{i,j} = \begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \leftarrow i \\ & & \mu & \ddots & \leftarrow j \\ & & & & 1 \end{bmatrix}$$

	determinant	Inverse
P_{ij}	1 if $i=j$ -1 if $i\neq j$	P_{ij}
$M_{i,\alpha}$	α	$M_{i,1/\alpha}$
$E^{\mu}_{i,j}$	1	$E^{-\mu}_{i,j}$

The great importance of working with elementary matrices stems from the fact that they allow the interpretation of an operation in terms of matrix multiplication.

Lemma Let B be an elementary matrix of one of the three types and let A be a given matrix.

(i) The matrix BA is the matrix obtained by applying to the rows of A the operation that corresponds to B.

(ii) The matrix AB^T is the matrix obtained by applying to the columns of A the operation that corresponds to B

proof. Exercise.

Remark It is clear now that naive Gaussian elimination is equivalent to multiplying $[A|b]$ or A from the left by a sequence of elementary matrices of type III.

Theorem Let the $n\times n$ real matrix A be such that the leading principal minors are all nonzero. Then, there exists a unit lower triangular matrix L and an upper triangular matrix U with $u_{ii}\neq 0$, $i=1,\dots,n$, such that $A=LU$. Furthermore, L and U are uniquely defined.

proof. We already saw that the conditions on A imply that naive Gaussian Elimination applied to A will not fail since no zero pivots were never encountered. So we have a sequence of elementary row operations of type III

leading to U. Equivalently, we have

$$U = E_{n,n-1}^{-\mu_{n,n-1}} E_{n,n-2}^{-\mu_{n,n-2}} E_{n-1,n-2}^{-\mu_{n-1,n-2}} \cdots E_{n,2}^{-\mu_{n,2}} \cdots E_{3,2}^{-\mu_{3,2}} E_{n,1}^{-\mu_{n,1}} \cdots E_{2,1}^{-\mu_{2,1}} A .$$

Note that we have $(-)$ in front of the multiplier μ since we use subtraction in the basic elimination step.

We have

$$A = E_{2,1}^{\mu_{2,1}} \cdots E_{n,1}^{\mu_{n,1}} \cdots\cdots E_{n,n-1}^{\mu_{n,n-1}} U \equiv L U .$$

L is the product of unit lower triangular matrices and hence is unit lower triangular. Furthermore, given the particular ordering of the elimination matrices in the sequence, it is easily shown that

$$L = \begin{bmatrix} 1 & & & \\ \mu_{21} & 1 & & 0 \\ \mu_{31} & \mu_{32} & 1 & \\ \vdots & \vdots & & \ddots \\ \mu_{n1} & \mu_{n2} & \cdots & \mu_{n,n-1} \quad 1 \end{bmatrix},$$

that is the multipliers μ occupy the same location in L in which they arose during the elimination process.

To prove that L and U are uniquely defined, suppose we have

$$A = L_1 U_1 = L_2 U_2 .$$

Then

$$L_2^{-1} L_1 = U_2 U_1^{-1} .$$

Now L_2^{-1} is also unit lower triangular and so is $L_2^{-1} L_1$. Also, U_1^{-1} is upper triangular and so is $U_2 U_1^{-1}$. Hence $L_2^{-1} L_1$ and $U_2 U_1^{-1}$ must both be diagonal. Furthermore for any $i = 1, \ldots, n$

$$(L_2^{-1} L)_{ii} = \sum_{j=1}^{n} (L_2^{-1})_{ij} (L_1)_{ji}$$

$$= \sum_{j=1}^{i-1} (L_2^{-1})_{ij} \underbrace{(L_1)_{ji}}_{0} + \underbrace{(L_2^{-1})_{ii}}_{1} \underbrace{(L_1)_{ii}}_{1} + \sum_{j=i+1}^{n} \underbrace{(L_2^{-1})_{ij}}_{0} (L_1)_{ji}$$

$= 1$. Thus $L_2^{-1} L_1 = I \Rightarrow L_2 = L_1$. Hence $U_2 U_1^{-1} = I \Rightarrow U_2 = U_1$. ∎

$A = LDL^T$ Factorization

Suppose the matrix A has all of its leading principal minors nonzero and is also symmetric. We know that naive Gaussian elimination will not fail and will lead to the factorization $A = LU$. Exploiting the symmetry of A we can show that in fact we have the factorization $A = LDL^T$ where L is unit lower triangular and D is diagonal with nonzero diagonal elements. Moreover, this can be accomplished with half the work required for general $A = LU$ factorization.

Theorem Let A be symmetric and have all its leading principal minors nonzero. Then there exists a unit lower triangular matrix L and a diagonal matrix D with nonzero diagonal elements such that $A = LDL^T$. Furthermore, L and D are uniquely defined.

proof.

Given that the leading principal minors are nonzero, there exists a unique unit lower triangular matrix L and unique upper triangular matrix U with nonzero diagonal elements such that $A = LU$.

Clearly we can write $U = D\tilde{U}$ where D is diagonal with $D_{ii} = u_{ii}$ and $\tilde{U}$ is upper triangular with diagonal elements equal to 1. Hence $\tilde{U} = \tilde{L}^T$ where $\tilde{L}$ is unit lower triangular. So far

$$A = LU = LD\tilde{L}^T.$$

we will show that $\tilde{L} = L$. Indeed, note that since A is symmetric

$$A = A^T = \tilde{L}DL^T = \tilde{L}\hat{U} \quad \text{with } \hat{U} \equiv DL^T.$$

we have shown that

$$A = LU = \tilde{L}\hat{U}.$$

Since the L, U are uniquely defined, we must have $\tilde{L} = L$. The same way, we can show that if $A = L_1 D_1 L_1^T = L_2 D_2 L_2^T$, then $L_2 = L_1$ and $D_2 = D_1$. ∎

There are algorithmic details which were not revealed by the proof above. Indeed, the $A = LDL^T$ factorization is

a modification of the $A = LU$ factorization and exploits the symmetry of A leading to a saving of half the work.

The essential fact used here is that post multiplication by an elementary matrix of type III has the same effect on columns as premultiplication has on rows. Consequently,

$$\circledast \quad A^{(2)} = \overbrace{E_{n,1}^{-\mu_{n,1}} E_{n-1,1}^{-\mu_{n-1,1}} \cdots E_{2,1}^{-\mu_{21}} A^{(1)}}^{M^{(1)}} \overbrace{\left(E_{2,1}^{-\mu_{21}}\right)^T \cdots \left(E_{n-1,1}^{-\mu_{n-1,1}}\right)^T \left(E_{n,1}^{-\mu_{n,1}}\right)^T}^{M^{(1)T}}$$

$$\equiv M^{(1)} A^{(1)} M^{(1)T} = \begin{bmatrix} a_{11}^{(1)} & 0 & \cdots & 0 \\ 0 & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} \\ \vdots & \vdots & & \vdots \\ 0 & a_{n2}^{(2)} & \cdots & a_{nn}^{(2)} \end{bmatrix} \equiv \left[\begin{array}{c|c} a_{11}^{(1)} & 0 \\ \hline 0 & B_{22} \end{array}\right]$$

we observe the following important facts

(i) $A^{(2)}$ is symmetric. Hence so is B_{22}.

(ii) $\circledast$ is a mathematical description of the algorithm. The algorithm itself does the pre multiplication by $M^{(1)}$.

(iii) The post multiplication by $(M^{(1)})^T$ is not part of the algorithm since we knows the outcome.

(iv) Since we know that B_{22} is symmetric, we need only update its lower triangular part. This requires

$2(1+2+\cdots+n-1) = (n-1)n$ operations vs. $2(n-1)^2$ in the general (nonsymmetric) case).

```
for i = 2, ..., n
    m_{i1} = a_{i1}^{(1)} / a_{11}^{(1)}
    for j = 2, ..., i
        a_{ij}^{(2)} = a_{ij}^{(1)} - m_{i1} a_{1j}^{(1)}
    }
}
```

vs.

```
for i = 2, ..., n
    m_{i1} = a_{i1}^{(1)} / a_{11}^{(1)}
    for j = 2, ..., n
        a_{ij}^{(2)} = a_{ij}^{(1)} - m_{i1} a_{1j}^{(1)}
    }
}
```

This is really the heart of the matter. Also note that there is no need to set $a_{12}^{(1)}, \ldots, a_{1n}^{(1)}$ to zero after using them to update $a_{ij}^{(2)}$, $2 \le j \le i \le n$.

proceeding in this manner, at the k-th step, we have, with the by now obvious notation

$$A^{(k+1)} = M^{(k)} A^{(k)} \left(M^{(k)}\right)^T = \begin{bmatrix} a_{11}^{(1)} & & & 0 \\ & \ddots & & \\ & & a_{kk}^{(k)} & 0 \\ & & 0 & a_{k+1,k+1}^{(k+1)} \\ & & & \vdots \\ 0 & & & a_{n,k+1}^{(k+1)} \end{bmatrix}.$$

Algorithm: $A = LDL^T$ factorization

A symmetric, all leading principal minors nonzero

for $k = 1, 2, \ldots, n-1$

  for $i = k+1, \ldots, n$

    $m_{ik} = a_{i,k}^{(k)} / a_{k,k}^{(k)}$

    for $j = k+1, \ldots i$

      $a_{ij}^{(k+1)} = a_{ij}^{(k)} - m_{ik} a_{jk}^{(k)}$ ← note this is a change from $a_{kj}^{(k)}$

    }

  }

}

OUTPUT

$$L = \begin{bmatrix} 1 & & & 0 \\ m_{21} & 1 & & \\ m_{31} & m_{32} & \ddots & \\ \vdots & \vdots & & 1 \\ m_{n1} & m_{n2} & m_{n,n-1} & \end{bmatrix}, \quad D = \begin{bmatrix} a_{11}^{(1)} & & & 0 \\ & a_{22}^{(2)} & & \\ & & \ddots & \\ 0 & & & a_{nn}^{(n)} \end{bmatrix}.$$

work $= \sum_{k=1}^{n-1} (n-k)(n-k+2) = \frac{1}{3} n^3 + O(n^2).$

Special Matrices: strictly diagonally dominant matrices

Defn. A square matrix A is said to be **strictly row diagonally dominant** if

$$|a_{ii}| > \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}|, \quad i = 1, \cdots, n.$$

Ex.
$$A = \begin{bmatrix} -9 & 4 & 3 \\ 5 & 7 & 1 \\ 10 & -9 & 20 \end{bmatrix}.$$

Similarly, A is strictly column diagonally dominant if A^T is strictly row diagonally dominant.

In general, Gaussian Elimination, naive or otherwise, can be implemented stably for such matrices. As a first result we have

Theorem. Let A be row or column strictly diagonally dominant. Then its leading principal minors are all nonzero.

proof Assume first that A is row strictly diagonally dominant. For any k, $1 \le k \le n$ let $A^{(k)}$ denote the k-th leading principal submatrix of A. We will show that $A^{(k)}$ is nonsingular. Suppose otherwise, i.e. $\exists\, x \in R^k$, $x \neq 0$ such that $A^{(k)}x = 0$. Now there is an integer $1 \le m \le k$ such that $|x_m| = \|x\|_\infty = \max_{1 \le j \le k} |x_j|$. Since $x \neq 0$, $x_m \neq 0$.

$$0 = A^{(k)}x \Rightarrow a_{mm}x_m = -\sum_{j=1, j\neq m}^{k} a_{mj}x_j$$

$$\Rightarrow |a_{mm}||x_m| \le \sum_{j=1, j\neq m}^{k} |a_{mj}||x_j| \le |x_m| \sum_{j=1, j\neq m}^{k} |a_{mj}|$$

Since $|x_m| \neq 0$

$$|a_{mm}| \le \sum_{j=1, j\neq m}^{k} |a_{mj}| \le \sum_{j=1, j\neq m}^{n} |a_{mj}| < |a_{mm}|$$

which is a contradiction of strict row diagonal dominance.

The proof of the case when A is column strictly diagonally dominant follows from the fact that the determinant of a matrix and its transpose are the same. ∎

Special matrices: Symmetric positive definite matrices

Defn. A real symmetric matrix is said to be (symmetric) **positive definite** if

$$x^T A x \geq 0, \quad \forall x \in \mathbb{R}^n \text{ and } x^T A x = 0 \Rightarrow x = 0.$$

Equivalently, if

$$x^T A x > 0, \quad \forall x \in \mathbb{R}^n, \ x \neq 0.$$

Ex. $A = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}$.

Let $x \in \mathbb{R}^3$. Some algebra reveals that

$$x^T A x = 2x_1^2 - 2x_1 x_2 + 2x_2^2 - 2x_2 x_3 + 2x_3^2$$

$$= x_1^2 + (x_1 - x_2)^2 + (x_2 - x_3)^2 + x_3^2$$

which shows that A is s.p.d.

Verifying that a given symmetric matrix is positive definite by using the definition is not as easy as the example above shows. Fortunately, there are many criteria which can be used.

Theorem. Let A be a symmetric, positive definite matrix. Then

(i) $a_{ii} > 0 \quad i = 1, \ldots, n$

(ii) $a_{ij}^2 < a_{ii} a_{jj} \quad \forall i \neq j.$

Proof (i) we have $e_i^T A e_i = a_{ii} > 0$ since $e_i \neq 0$, $i = 1, \ldots, n$.

(ii) Let $i \neq j$ and let $x = e_i + t e_j \neq 0 \ \forall t$. Now $\forall t$

$$0 < x^T A x = (e_i + t e_j)^T A (e_i + t e_j) = a_{ii} + t a_{ji} + t a_{ij} + t^2 a_{jj}$$

$$= t^2 a_{jj} + 2t a_{ij} + a_{ii}$$

since A is symmetric.

The quadratic polynomial $p(t) = t^2 a_{jj} + 2a_{ij}t + a_{ii}$ is positive for all t; hence it does not have real roots. This can happen only if the discriminant $4(a_{ij})^2 - 4a_{jj}a_{ii}$ is negative. This is the required result. ∎

Remark In particular, it follows from (ii) of the last result that the largest element of an s.p.d. matrix must be located on its diagonal.

Theorem Let A be a symmetric matrix. The following statements are equivalent.

(a) A is positive definite.
(b) the leading principal minors are all positive.
(c) the eigenvalues of A (which are real since A is symmetric) are positive.
(d) $A = LDL^T$ where L is unit lower triangular and D is diagonal with positive diagonal elements.
(e) $A = LL^T$ where L is lower triangular with nonzero diagonal elements.

Proof
we shall first prove (a)⟺(c) and then follow the pattern of implications (a) ⇒ (b) ⇒ (d) ⇒ (e) ⇒ (a).

(a) ⇒ (c) Let λ be an eigenvalue of A with corresponding eigenvector $v \neq 0$. Note that both λ and v are real. We have

$$Av = \lambda v \Rightarrow v^T A v = v^T(\lambda v) = \lambda \|v\|_2^2.$$

Since $v^T A v > 0$, we must have $\lambda > 0$.

(c) ⇒ (a). A is symmetric, hence there exists a real orthogonal matrix Q ($Q^{-1} = Q^T$) such that $Q^T A Q = \Lambda = \text{diag}\{\lambda_1, \ldots, \lambda_n\}$.

$$\Rightarrow A = Q\Lambda Q^T \Rightarrow x^T A x = x^T Q \Lambda Q^T x = y^T \Lambda y, \quad y = Q^T x.$$

Since Λ is diagonal, $x^T A x = \sum_{i=1}^{n} \lambda_i y_i^2$.

Now for $x \neq 0$, $y = Q^T x$ is nonzero, hence $x^T A x = \sum_{i=1}^{n} \lambda_i y_i^2 > 0$.

In particular, we have shown that if A is s.p.d., then $\circledast$ $\det(A) = \lambda_1 \cdot \lambda_2 \cdots \lambda_n > 0$.

(a) $\Rightarrow$ (b).

We shall prove the following: If A is s.p.d., then so are all of its leading principal submatrices. Indeed, let

$$A = \left[\begin{array}{c|c} A_{11} & A_{12}^T \\ \hline A_{12} & A_{22} \end{array}\right], \quad A_{11} \text{ is the } k \times k \text{ leading principal submatrix of } A \text{ and } x = \begin{bmatrix} x_1 \\ \vdots \\ x_k \\ \hline 0 \end{bmatrix} = \begin{bmatrix} w \\ \hline 0 \end{bmatrix}, \; w \in R^k$$

we have

$x^T A x = w^T A_{11} w$, also $x \neq 0 \Leftrightarrow w \neq 0$. Thus A s.p.d. $\Rightarrow$ A_{11} s.p.d. From $\circledast$, we have $\det(A_{11}) > 0$.

(b) $\Rightarrow$ (d) The leading principal minors being positive, are nonzero; hence by a previously established result, we have the $A = LDL^T$ factorization of A with $D_{ii} \neq 0$. We need to show that in fact $D_{ii} > 0$. To see this, note that for any $i = 1, \ldots, n$, there exists $x_i \neq 0$ such that $L^T x_i = e_i$, this being true since L is invertible. Hence

$$0 < x_i^T A x_i = x_i^T L D L^T x_i = e_i^T D e_i = D_{ii}.$$

(d) $\Rightarrow$ (e). Let $D^{1/2} = \operatorname{diag}\{D_{11}^{1/2}, D_{22}^{1/2}, \ldots, D_{nn}^{1/2}\}$.

$$A = LDL^T = LD^{1/2}D^{1/2}L^T = (LD^{1/2})(LD^{1/2})^T = \tilde{L}\tilde{L}^T.$$

Clearly $\tilde{L}_{ii} = D_{ii}^{1/2} \neq 0$. Removing the tilde gives the result.

(e) $\Rightarrow$ (a)

Let $x \neq 0$. We have $x^T A x = x^T L L^T x = y^T y$, $y \equiv L^T x$.

Hence

$$x^T A x = \sum_{i=1}^{n} y_i^2. \quad \text{Clearly } x \neq 0 \Leftrightarrow y \neq 0.$$

Hence

$x^T A x > 0$. This completes the proof of the Theorem. ∎

Cholesky Factorization for symmetric positive definite matrices

By a previous result we saw that a symmetric matrix A is positive definite if and only if $A = \tilde{L} D \tilde{L}^T$ where $\tilde{L}$ is unit lower triangular and D is a diagonal matrix with positive diagonal elements. From this we easily obtain the factorization $A = L L^T$ with $L = \tilde{L} D^{1/2}$.

It turns out that there is a different, "more direct," way of calculating L. This is an example of a <u>direct factorization</u> approach in contrast to elimination.

The algorithm starts by writing

$$\begin{bmatrix} l_{11} & & & 0 \\ l_{21} & l_{22} & & \\ \vdots & & \ddots & \\ l_{n1} & \cdots & & l_{nn} \end{bmatrix} \begin{bmatrix} l_{11} & l_{21} & \cdots & l_{n1} \\ & l_{22} & & \vdots \\ 0 & & \ddots & \\ & & & l_{nn} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$$

and then by implementing a sequence of operations of the form

$$(\text{row } i \text{ of } L) \cdot (\text{col } j \text{ of } L^T) = a_{ij}$$

to calculate the elements l_{ij} of L. Note that there are different sequences for achieving this. The following algorithm is one such where L is calculated one column at a time, starting at the top of the column.

Algorithm: Cholesky

$l_{11} = \sqrt{a_{11}}$

for $i = 2, \cdots, n$

  $l_{j1} = a_{j1} / l_{11}$

}

for $i = 2, \cdots, n-1$

  $l_{ii} = \left(a_{ii} - \sum_{k=1}^{n-1} l_{ik}^2\right)^{1/2}$

  for $j = i+1, \cdots, n$

    $l_{ji} = \left(a_{ji} - \sum_{k=1}^{i-1} l_{jk} l_{ik}\right) / l_{ii}$

  }

}

$l_{nn} = \left(a_{nn} - \sum_{k=1}^{n-1} l_{nk}^2\right)^{1/2}$.

Ex.

$$A=\begin{bmatrix}3 & -3 & 6\\ -3 & 7 & -7\\ 6 & -7 & 13\end{bmatrix}=\begin{bmatrix}l_{11} & 0 & 0\\ l_{21} & l_{22} & 0\\ l_{31} & l_{32} & l_{33}\end{bmatrix}\begin{bmatrix}l_{11} & l_{21} & l_{31}\\ 0 & l_{22} & l_{32}\\ 0 & 0 & l_{33}\end{bmatrix}$$

row 1 · col 1 $= a_{11} \Rightarrow l_{11}^2 = 3 \Rightarrow l_{11} = \sqrt{3}$. We choose positive root even though this is not necessary

row 2 · col 1 $= a_{21} \Rightarrow l_{21} l_{11} = -3 \Rightarrow l_{21} = -\sqrt{3}$

row 3 · col 1 $= a_{31} \Rightarrow l_{31} l_{11} = 6 \Rightarrow l_{31} = 2\sqrt{3}$

row 2 · col 2 $= a_{22} \Rightarrow l_{21}^2 + l_{22}^2 = 7 \Rightarrow l_{22} = \sqrt{7-(-\sqrt{3})^2} = 2$

row 3 · col 2 $= a_{32} \Rightarrow l_{31} l_{21} + l_{32} l_{22} = -7 \Rightarrow l_{32} = (-7-(2\sqrt{3})(-\sqrt{3}))/2 = -\frac{1}{2}$

row 3 · col 3 $= a_{33} \Rightarrow l_{31}^2 + l_{32}^2 + l_{33}^2 = 13 \Rightarrow l_{33} = \sqrt{13-(2\sqrt{3})^2-(-\frac{1}{2})^2} = \frac{\sqrt{3}}{2}$.

Remark we could have equally well modified this sequence to one where elements of L are computed row by row, i.e. we compute these according to the sequence

$$l_{11}, l_{21}, l_{22}, l_{31}, l_{32}, l_{33}.$$

The choice of a particular sequence reflects the characteristics of the computer e.g. parallelism.

Gaussian Elimination with pivoting strategies

An example we provided showed that naive Gaussian Elimination can fail even when using exact arithmetic. The problem is even worse in the presence of roundoff errors. In fact, the condition that all the leading principal minors be nonzero is a rather strong condition and it is quite "risky" to assume that it holds in a particular situation. In conclusion, naive Gaussian elimination must be modified in ways to make it much more robust (trustworthy).

Fortunately, it turns out that there are simple strategies that can considerably enhance the robustness of Gaussian Elimination. Before embarking on a study of some commonly used approaches, we shall consider an example

Ex, Solve the system $\begin{bmatrix} .003 & 59.14 \\ 5.291 & -6.130 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 59.17 \\ 46.78 \end{bmatrix}$

using 4-decimal digit arithmetic with rounding. Note that the solution is $x = 10,\ y = 1$.

$$fl(m_{21}) = fl\left(\frac{5.291}{.003}\right) = fl(1763.\overline{6}) = 1764.$$

In exact arithmetic, naive Gaussian Elimination gives

$$\left[A^{(2)} \vdots b^{(2)}\right] = \left[\begin{array}{cc:c} .003 & 59.14 & 59.17 \\ 0 & -104309.376\overline{} & -104309.376\overline{} \end{array}\right]$$

However, the calculated augmented matrix in the precision used is

$$fl\left[A^{(2)} \vdots b^{(2)}\right] = \left[\begin{array}{cc:c} .003 & 59.14 & 59.17 \\ 0 & -104300 & -104400 \end{array}\right].$$

Indeed

$$fl(a_{22}^{(2)}) = fl(-6.130 - 1764 * 59.14) = fl(-6.130 - \overbrace{fl(1764 * 59.14)}^{104322.96})$$

$$= fl(-6.130 - 104300) = fl(104306.13) = -104300.$$

$$fl(b_2^{(2)}) = fl(46.78 - 1764 * 59.17) = fl(46.78 - \underbrace{fl(1764 * 59.17)}_{104375.88})$$

$$= fl(46.78 - 104400) = fl(-104353.22) = -104400.$$

Hence

$$fl(y) = fl\left(\frac{-104400}{-104300}\right) = fl(1.00095\overline{87}\ldots) = 1.001$$

Not too bad as an approximation of y!

It's too early to declare victory, however. Indeed,

$$fl(x) = fl\left(\frac{59.17 - fl(59.14 * \overbrace{fl(y)}^{1.001})}{.003}\right) = fl\left(\frac{59.17 - fl(59.19914)}{.003}\right)$$

$$= fl\left(\frac{59.17 - 59.20}{.003}\right) = fl\left(\frac{-.03}{.003}\right) = fl(-10) = -10.!!!$$

How do we explain this? The leading principal minors are $.003$ and -312.93, both nonzero and naive Gaussian elimination did not fail, even in finite precision.

Is this a case of an ill-conditioned matrix? Not really since $\text{cond}_\infty(A) = 12.3$.

It appears therefore that the algorithm itself was the cause of the large errors and in particular the large multiplier (1764) encountered. This suggests redoing the problem with the rows interchanged before the elimination process begins

$$\left[\begin{array}{cc|c} 5.291 & -6.130 & 46.78 \\ .003 & 59.14 & 59.17 \end{array}\right]$$

Now

$$fl(m_{21}) = fl\left(\frac{.003}{5.291}\right) = fl(5.67\,00567 \times 10^{-4}) = 5.670 \times 10^{-4}$$

$$fl(a_{22}^{(2)}) = fl(59.14 - fl(\overbrace{5.670\times10^{-4}\times(-6.130)}^{-.0034757 1}))$$
$$= fl(59.14 + .0035) = fl(59.1435) = 59.14.$$

$$fl(b_2^{(2)}) = fl(59.17 - fl(\underbrace{5.670\times10^{-4}\times 46.78}_{.02652426}))$$
$$= fl(59.17 - .02652) = fl(59.14348) = 59.14$$

Thus,

$$fl(y) = fl(59.14/59.14) = 1$$

$$fl(x) = fl\left(\frac{46.78 + fl(6.130\times \overbrace{fl(y)}^{=1})}{5.291}\right) = fl\left(\frac{\overbrace{46.78+6.130}^{52.91}}{5.291}\right)$$
$$= fl\left(\frac{52.91}{5.291}\right) = fl(10) = 10.$$

The fact that we recovered the exact values is just a coincidence. In general, we can only expect improvement in the result.

Remark It is important to keep in mind that the row interchanges and other similar strategies are designed to improve the performance of the algorithm and are not a cure of ill-conditioning.

Maximal row pivoting or partial pivoting

At the beginning of Gaussian Elimination, even if $a_{11}^{(1)}$ is nonzero, we scan column 1 and find the index p such that

$$|a_{p1}^{(1)}| = \max_{1\le i\le n} |a_{i1}^{(1)}|,$$

i.e. the largest element in absolute value of the first column. In case of a tie, we choose as p the first such occurrence. We then interchange rows 1 and p and use $a_{p1}^{(1)}$ as the pivot.

Proceeding this way, at the kth step, we have

$$[A^{(k)}, b^{(k)}] = \begin{bmatrix} a_{11}^{(1)} & \cdots & a_{1n}^{(1)} & b_1^{(1)} \\ & \ddots & & \\ & a_{kk}^{(k)} \cdots a_{kn}^{(k)} & & b_k^{(k)} \\ 0 & & & \\ & a_{nk}^{(k)} \cdots a_{nn}^{(k)} & & b_n^{(k)} \end{bmatrix}$$

we choose as pivot the element $a_{pk}^{(k)}$ such that

$$|a_{pk}^{(k)}| = \max_{k \le i \le n} |a_{ik}^{(k)}|,$$

and interchange rows p and k and proceed with the elimination.

Remarks (i) It is clear that as a result of this strategy, all the multipliers satisfy $|m_{ik}| \le 1$.

(ii) In actual implementations, we do not "physically" shuffle rows p and k in memory. Rather, we keep account of the interchanges by "indirect addressing" by means of what's called a pivot vector.

Ex.

$$[A^{(1)}, b^{(1)}] = \begin{bmatrix} 3 & -13 & 9 & 3 & -19 \\ -6 & 4 & 1 & -18 & -24 \\ 6 & -2 & 2 & 4 & 16 \\ 12 & -8 & 6 & 10 & 26 \end{bmatrix}; \quad PIV = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} \rightarrow PIV = \begin{bmatrix} 4 \\ 2 \\ 3 \\ 1 \end{bmatrix}$$

At $k=1$ we choose as pivot row $p=4$ and (12) as pivot.

$$\Rightarrow [A^{(2)}, b^{(2)}] = \begin{bmatrix} 0 & -11 & 15/2 & 1/2 & -25.5 \\ 0 & 0 & 4 & -13 & -11 \\ 9 & 2 & -1 & -1 & 3 \\ 12 & -8 & 6 & 10 & 26 \end{bmatrix}; \quad PIV = \begin{bmatrix} 4 \\ 2 \\ 3 \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} 4 \\ 1 \\ 3 \\ 2 \end{bmatrix}$$

For the second column, the pivot is (−11) which is on row 1

$$[A^{(3)}, b^{(3)}] = \left[\begin{array}{cccc|c} 0 & -11 & 15/2 & 1/2 & -51/2 \\ 0 & 0 & \textcircled{4} & -13 & -11 \\ 0 & 0 & 4/11 & -\frac{10}{11} & -\frac{18}{11} \\ 12 & -8 & 6 & 10 & 26 \end{array}\right]; \quad PIV = \begin{bmatrix} 4 \\ 1 \\ 3 \\ 2 \end{bmatrix} \rightarrow PIV = \begin{bmatrix} 4 \\ 1 \\ 2 \\ 3 \end{bmatrix}$$

For the 3rd column the pivot is (4) which is on row 2

$$[A^{(4)}, b^{(4)}] = \left[\begin{array}{cccc|c} 0 & -11 & 15/2 & 1/2 & -\frac{51}{2} \\ 0 & 0 & 4 & -13 & -11 \\ 0 & 0 & 0 & 3/11 & -7/11 \\ 12 & -8 & 6 & 10 & 26 \end{array}\right] \cdot \quad PIV = \begin{bmatrix} 4 \\ 1 \\ 2 \\ 3 \end{bmatrix}$$

Final pivot vector.

Note that $B(i,j) = A^{(4)}(piv(i), j)$ is the matrix A after elimination in upper triangular form.

Algorithm: Gaussian Elimination with partial pivoting

```
for i=1:n
   piv(i) = i
}

for k=1:n-1
   rowmax = piv(k)
   for i=k+1:n
      if abs(a(rowmax), k) < abs(a(piv(i), k))
         rowmax = piv(i)
         temp = i
      }
   }
   piv(temp) = piv(K)
   piv(k) = rowmax

   for i=k+1:n
      mult = a(piv(i), k) / a(piv(k), k)
      for j=k:n
         a(piv(i), j) = a(piv(i), j) - mult * a(piv(k), j)
      }
```

In practice, this should be j=k+1:n the cases j=k introduce the zeros in the lower triangular part of A.

```
        b(piv(i)) = b(piv(i)) - mult * b(piv(k))
    }
}
```

The corresponding back substitution is given by

Algorithm: Back substitution (to accompany Gaussian Elimination with partial pivoting)

```
x(n) = b(piv(n)) / a(piv(n), n)

for i = n-1 : 1
    row = piv(i)
    sum = 0
    for j = i+1 : n
        sum = sum + a(row, j) * x(j)
    }
    x(i) = (b(row) - sum) / a(row, i)
}
```

solution $x = (6.0556, -4.833, -10.333, -2.333)$.

Gaussian Elimination with scaled partial pivoting

we saw earlier that one of the benefits of partial (or maximal row) pivoting is to ensure that the multipliers are always bounded by 1 in absolute value.

Another way of looking at this is to realize that what really matters is the size of the potential pivot relative to the elements that are to its right in the same row.

Indeed, let's look at the basic elimination operation:

$$a_{ij} = a_{ij} - mult * a_{kj} = a_{ij} - \frac{a_{ik}}{a_{kk}} a_{kj}.$$

So instead of looking at the size of $\frac{a_{ik}}{a_{kk}}$, look at the size of a_{kj}/a_{kk} in the choice of pivot.

we compute scale factors $s_1, \ldots, s_n$

$$s_i = \max_{1 \le j \le n} |a_{ij}|, \quad i = 1, \ldots, n,$$

i.e. s_i is the largest element in absolute value in row i.

At step k of the elimination process, we choose as pivot the element $a_{pk}^{(k)}$ from the vector $[a_{kk}^{(k)}, \ldots, a_{nk}^{(k)}]^T$ such that

$$|a_{pk}^{(k)}| = \max_{k \le i \le n} |a_{ik}^{(k)}|.$$

Ex.

$$[A\,|\,b] = \left[\begin{array}{cccc|c} 3 & -13 & 9 & 3 & -19 \\ -6 & 4 & 1 & -18 & -24 \\ \textcircled{6} & -2 & 2 & 4 & 16 \\ 12 & -8 & 6 & 10 & 26 \end{array}\right]; \quad piv = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} \rightarrow \begin{bmatrix} 3 \\ 2 \\ 1 \\ 4 \end{bmatrix}$$

The scale factors are $\{13, 18, 6, 12\}$. we choose as pivot 6 since we have

$$\frac{|6|}{6} = \max\left\{\frac{|3|}{13}, \frac{|-6|}{18}, \frac{|6|}{6}, \frac{|12|}{12}\right\}$$

and this is the first occurrence of the maximum.

$$\left[A^{(2)} \,;\, b^{(2)}\right] = \left[\begin{array}{cccc|c} 0 & \boxed{-12} & 8 & 1 & -27 \\ 0 & 2 & 3 & -14 & -8 \\ 6 & -2 & 2 & 4 & 16 \\ 0 & \boxed{-4} & 2 & 2 & -6 \end{array}\right];$$

$$\frac{|-12|}{13} = \max\left\{\frac{|-12|}{13}, \frac{|2|}{18}, \frac{|-4|}{12}\right\}$$

The new pivot row is 1 $\Rightarrow$ $piv = \begin{bmatrix} 3 \\ 1 \\ 2 \\ 4 \end{bmatrix} \leftarrow \begin{bmatrix} 3 \\ 2 \\ 1 \\ 4 \end{bmatrix}$

$$\left[A^{(3)}, b^{(3)}\right] = \left[\begin{array}{cccc|c} 0 & -12 & 8 & 1 & -27 \\ 0 & 0 & \boxed{13/3} & -\frac{83}{6} & -25/2 \\ 6 & -2 & 2 & 4 & 16 \\ 0 & 0 & \boxed{-2/3} & 5/3 & 3 \end{array}\right]$$

$$\frac{13/3}{18} = \max\left\{\frac{13/3}{18}, \frac{|-2/3|}{12}\right\}$$

$piv = \begin{bmatrix} 3 \\ 1 \\ 4 \\ 2 \end{bmatrix} \leftarrow \begin{bmatrix} 3 \\ 1 \\ 2 \\ 4 \end{bmatrix}$

$\Rightarrow$ pivot row is 2 with pivot $\frac{13}{3}$.

$$\left[A^{(4)}, b^{(4)}\right] = \left[\begin{array}{cccc|c} 0 & -12 & 8 & 1 & -27 \\ 0 & 0 & 13/3 & -\frac{83}{6} & -25/2 \\ 6 & -2 & 2 & 4 & 16 \\ 0 & 0 & 0 & -\frac{6}{13} & 14/13 \end{array}\right]$$

solution is $x = (6.0556, -4.833, -10.333, -2.333)$, the same as the one obtained from Gaussian Elimination with partial pivoting.

Remark In general scaled row pivoting leads to a more stable algorithm, however it is costlier than partial pivoting. Coupled with the fact that partial pivoting performs quite well in most situations, explains why it is not used often.

PA = LU Factorization

We already saw examples of incorporating row interchanges into Gaussian elimination. This can be formalized as the PA = LU factorization where P is a permutation matrix and encapsulates the row interchanges performed.

For $1 \le i, j \le n$, let P_{ij} denote the <u>elementary permutation matrix</u> obtained by interchanging rows i and j of the identity matrix. We have

$$P_{ij} = P_{ij}^T = P_{ij}^{-1}$$

i.e. an elementary permutation matrix is symmetric and is its own inverse. More generally, we have

<u>Defn.</u> A <u>permutation matrix</u> P is the product of elementary permutation matrices.

<u>Ex.</u>

$$P = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{P_{34}} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}}_{P_{24}} \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{P_{12}} = P_{34} P_{24} P_{12}$$

Note that such (non elementary) permutation matrices are not symmetric in general. However it is easily verified that

$$P^T = P^{-1}, \text{ i.e. } P \text{ is } \underline{\text{orthogonal}}.$$

also that and this is important in applications such as Gaussian elimination that the information in P can be represented by a vector. In the above example

$$P = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \longleftrightarrow \begin{bmatrix} 2 \\ 4 \\ 1 \\ 3 \end{bmatrix} \longleftrightarrow \begin{bmatrix} e_2 \\ e_4 \\ e_1 \\ e_3 \end{bmatrix}$$

Theorem (PA=LU Factorization) Let A be an $n \times n$, real, invertible matrix. Then there exist a permutation matrix P, a unit lower triangular matrix L and an upper triangular matrix U with nonzero diagonal elements such that $PA = LU$. Furthermore, for a given P, the factors L and U are uniquely defined.

proof. We use induction on the size of A. The result is certainly true for 1×1 matrices.

Suppose A is 2×2

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}.$$

The fact that the result is true for 1×1 matrices is enough for starting the induction argument. We do the 2×2 case as an illustration of the overall argument

Since A is invertible, a_{11} and a_{21} cannot be both zero. If $a_{11} \neq 0$, then we use it as pivot to arrive at

$$A = \begin{bmatrix} 1 & 0 \\ \mu_{21} & 1 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} \\ 0 & u_{22} \end{bmatrix}, \quad \mu_{21} = \frac{a_{21}}{a_{11}}, \quad u_{22} = a_{22} - \mu_{21} a_{12}.$$

Clearly $u_{22} \neq 0$ since $\det(A) = a_{11} u_{22} \neq 0$.

On the other hand, if $a_{11} = 0$, then

$$P_{12} A = \begin{bmatrix} a_{21} & a_{22} \\ a_{11} & a_{12} \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 \\ \mu_{21} & 1 \end{bmatrix}}_{L} \underbrace{\begin{bmatrix} a_{21} & a_{22} \\ 0 & u_{22} \end{bmatrix}}_{U}, \quad \mu_{21} = a_{11}/a_{21}, \quad u_{22} = a_{12} - \mu_{21} a_{22}$$

Again, we must have $u_{22} \neq 0$.

Now assume that the result is true for all $(n-1) \times (n-1)$ real invertible matrices.

Since A is invertible, there must be a nonzero entry in its first column, say $a_{i_1,1}$, $1 \leq i_1 \leq n$. We permute rows 1 and i_1 (if $i_1 = 1$ no permution is done). We then use $a_{i_1,1}$ as pivot to do elimination on the first column. In other words,

$$\underbrace{E_{n,1}^{-\mu_{n,1}} \cdots E_{2,1}^{-\mu_{2,1}} P_{i_1,1}}_{E^{(1)}} A \equiv A^{(2)} = \begin{bmatrix} a_{i_1,1} & w^T \\ 0 & B \end{bmatrix}, \quad w = (a_{i_1,2}, \ldots, a_{i_1,n})$$

B $(n-1) \times (n-1)$ matrix.

The matrix $A^{(2)}$ is invertible being the product of invertible matrices. Also, in view of the block structure of $A^{(2)}$,

$$0 \neq \det(A^{(2)}) = a_{1,1}^{(1)} \det(B).$$

Hence B is invertible. By the induction hypothesis, there exist a permutation matrix $\tilde{P}$, a unit lower triangular matrix $\tilde{L}$ and an upper triangular matrix with nonzero diagonal elements (all 3 matrices are $(n-1)$ by $(n-1)$) such that $\tilde{P}B = \tilde{L}\tilde{U}$. Thus, using block multiplication

$$E^{(1)}P_{1,1}A = \left[\begin{array}{c|c} a_{1,1}^{(1)} & w^T \\ \hline 0 & \tilde{P}^T\tilde{L}\tilde{U} \end{array}\right] = \left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & \tilde{P}^T \end{array}\right]\left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & \tilde{L} \end{array}\right]\left[\begin{array}{c|c} a_{1,1}^{(1)} & w^T \\ \hline 0 & \tilde{U} \end{array}\right] \equiv \hat{P}^T\hat{L}\hat{U}.$$

Note that $\hat{P}$ is a permutation matrix, $\hat{L}$ is unit lower triangular and $\hat{U}$ is upper triangular with nonzero diagonal elements. From the last equality, we obtain

(*) $$\hat{P}E^{(1)}P_{1,1}A = \hat{L}\hat{U}.$$

The matrices $\hat{P}$ and $E^{(1)}$ do not commute. However they interact in a very special way as we next show.

Let $\mu = (-\mu_{2,1}, -\mu_{3,1}, \ldots, -\mu_{n,1})^T \in R^{n-1}$ be the vector of multipliers of the first column that appear in $E^{(1)}$.

$$\hat{P}E^{(1)} = \left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & \tilde{P} \end{array}\right]\left[\begin{array}{c|c} 1 & 0 \\ \hline \mu & I_{n-1} \end{array}\right] = \left[\begin{array}{c|c} 1 & 0 \\ \hline \tilde{P}\mu & \tilde{P} \end{array}\right]$$

$$= \left[\begin{array}{c|c} 1 & 0 \\ \hline \tilde{P}\mu & I_{n-1} \end{array}\right]\left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & \tilde{P} \end{array}\right] \equiv \tilde{E}^{(1)}\hat{P}.$$

Note that $\tilde{E}^{(1)}$ is unit lower triangular and is

obtained from $E^{(1)}$ by permuting the column of multipliers. Using the latter equality in (*), we get

$$\tilde{E}^{(1)} \hat{P} P_{i_1,1} A = \hat{L}\hat{U} \Rightarrow (\hat{P} P_{i_1,1}) A = (\tilde{E}^{(1)})^{-1} \hat{L}\ \hat{U}.$$

Observe the following facts: $\hat{P} P_{i_1,1}$ is a permutation matrix; call it P. $(\tilde{E}^{(1)})^{-1}$ is unit lower triangular and so is $(\tilde{E}^{(1)})^{-1}\hat{L}$; call that L. Finally with $U = \hat{U}$, we have $PA = LU$ and the proof is complete. ∎

Remark If P, L and U are available, the system $Ax = b$ can be solved as follows:

$$Ax = b \Leftrightarrow PAx = Pb \Leftrightarrow LUx = P^T b$$

(i) Compute $P^T b$. We do not perform this as matrix-vector multiplication but rather using the vector v that represents P.

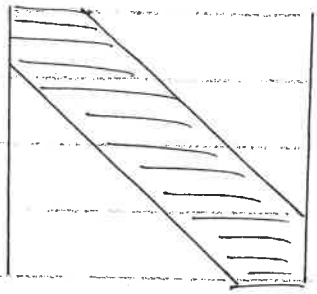
(ii) Solve for y from $Ly = P^T b$ using Forward substitution

(iii) Solve for x from $Ux = y$ using Back substitution.

§6.3 Tridiagonal and banded systems

Defn A matrix A is called **banded** if there is an integer k (typically $k \ll n$) such that

$$a_{ij} = 0 \quad \text{if} \quad |i-j| \geq k$$



$k=1 \Rightarrow$ diagonal

$k=2 \Rightarrow$ **tridiagonal** : diagonal + subdiagonal + superdiagonal

$k=3 \Rightarrow$ **pentadiagonal** : Total of 5 diagonals including main diagonal

In general, the number of nonzero diagonals is $\boxed{2k-1}$

Also in cubic spline interpolation

- banded matrices arise from solving differential equations
- only the nonzero elements need be stored : much less than n^2
- Gaussian elimination can be adapted to exploit banded structure. Especially if pivoting is not applied.

Tridiagonal systems

Gaussian Elimination, without pivoting, applied to the tridiagonal system

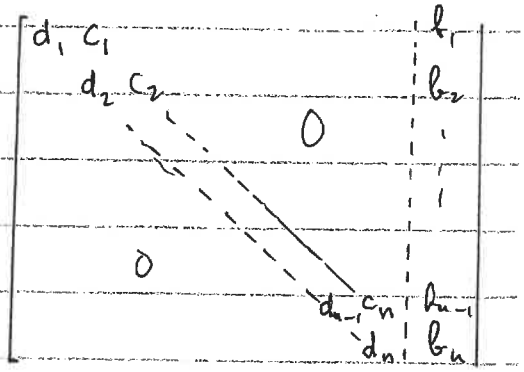
$$\begin{bmatrix} d_1 & c_1 & & & & b_1 \\ a_1 & d_2 & c_2 & & 0 & b_2 \\ & a_2 & \ddots & \ddots & & \vdots \\ & & \ddots & \ddots & c_{n-1} & \\ & 0 & & a_{n-1} & d_n & b_n \end{bmatrix}$$

$$d_i \leftarrow d_i - \frac{a_{i-1}}{d_{i-1}} c_{i-1} \qquad i = 2, \ldots, n$$

$$b_i \leftarrow b_i - \frac{a_{i-1}}{d_{i-1}} b_{i-1} \qquad i = 2, \ldots, n$$

This gives the triangular system where $d_2, \ldots, d_n$ and $b_2, \ldots, b_n$ have been modified

Then we can solve for $x_n, \ldots, x_1$ by back substitution



$$x_n = \frac{b_n}{d_n}, \qquad x_i = \frac{b_i - c_{i+1} x_{i+1}}{d_i}, \quad i = n-1, n-2, \ldots, 1.$$

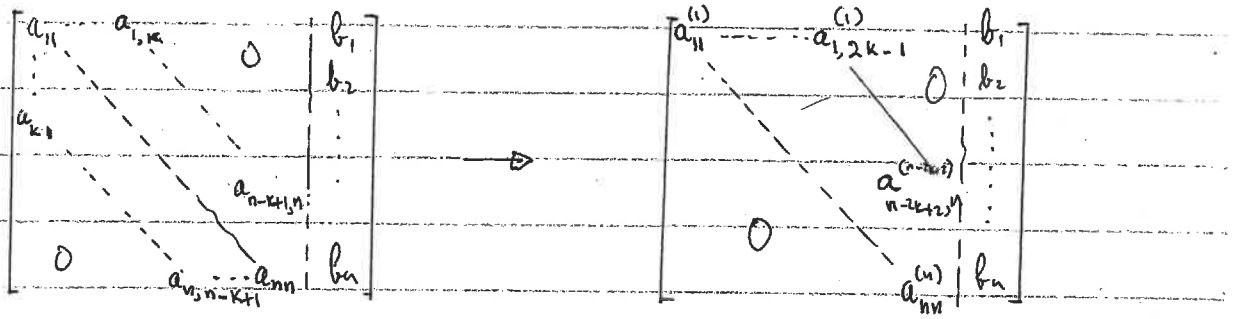
Total cost: $8n$

The cost would have been $\approx \frac{2}{3} n^3$ if the system was full!

Remarks

1) Similar algorithms exist for more general banded systems

2) pivoting can be incorporated (This is a must in most situations) without much change in the storage and efficiency characteristics of the algorithms



Question For what classes of matrices is pivoting unnecessary, i.e. (i) no zero pivots will be encountered

(ii) pivoting will not enhance stability

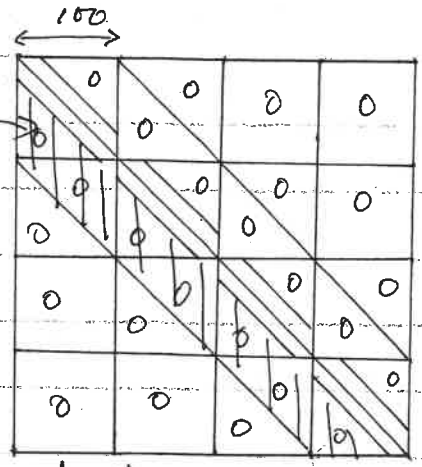
§6.5 Iterative Solution of Linear systems

When a matrix A is large ($n >> 1000$) and sparse i.e. consisting mainly of zeros eg. having at most 5 or 10 nonzeros on each row, Gaussian Elimination can be quite inefficient in run time and may require huge amounts of storage due to fill in.

small
each block is 100×100

and there are 100×100 blocks fill in

$\Rightarrow n = 10,000$



Original matrix has 5 nonzeros/row

after GE is done, the area between the inner & outer bands will be filled in. $\Rightarrow$ need $\approx 10,000 \times 100$ words of storage.

~~A general iterative method for solving $Ax=b$ goes as follows:~~
~~Select a nonsingular matrix Q and having chosen an arbitrary initial vector~~

Example of an iterative method: Jacobi's method

Consider the system $Ax=b$, which we write in comp. form

$$a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1$$
$$a_{21}x_2 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2$$
$$\vdots \qquad\qquad \vdots \qquad\qquad \vdots$$
$$a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n$$

All iterative methods require (like Newton's method) a starting vector $x^{(0)}$. The idea is to generate a sequence $x^{(1)}, x^{(2)}, \ldots, x^{(k)}, \ldots$ that will converge to the solution x. Of course, we have to stop the iteration at some point.

In Jacobi's method, we use the i-th row to calculate a new, and hopefully improved value for x_i as follows:

$$a_{11}x_1^{(1)} + a_{12}x_2^{(0)} + \cdots + a_{1n}x_n^{(0)} = b_1 \Rightarrow x_1^{(1)} = \frac{b_1 - \sum_{j=1,\, j\neq 1}^{n} a_{1j}x_j^{(0)}}{a_{11}}$$

$$a_{21}x_1^{(0)} + a_{22}x_2^{(1)} + a_{23}x_3^{(0)} + \cdots + a_{2n}x_n^{(0)} = b_2 \Rightarrow x_2^{(1)} = \frac{b_2 - \sum_{j=1,\, j\neq 2}^{n} a_{2j}x_j^{(0)}}{a_{22}}$$

$$\vdots$$

$$a_{n1}x_1^{(0)} + a_{n2}x_2^{(0)} + \cdots + a_{n,n-1}x_{n-1}^{(0)} + a_{nn}x_n^{(1)} = b_n \Rightarrow x_n^{(1)} = \frac{b_n - \sum_{j=1,\, j\neq n}^{n} a_{nj}x_j^{(0)}}{a_{nn}}$$

Remark Jacobi's method requires that $a_{ii} \neq 0$, $i = 1, \ldots, n$.

In general, given $x^{(k)}$, $x^{(k+1)}$ can be obtained (from $x^{(k)}$) exactly as $x^{(1)}$ was obtained from $x^{(0)}$:

$$x_i^{(k+1)} = \frac{b_i - \sum_{j=1,\, j\neq i}^{n} a_{ij}x_j^{(k)}}{a_{ii}}, \quad i = 1, \ldots, n$$

$x^{(k)} \to x^{(k+1)}$ is one iteration.

Ex.

let $A = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 3 & -1 \\ 0 & -1 & 2 \end{bmatrix}$, $b = \begin{bmatrix} 1 \\ 8 \\ -5 \end{bmatrix}$.

Suppose $x^{(0)} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$.

$$x_1^{(1)} = \frac{b_1 - a_{12} x_2^{(0)} - a_{13} x_3^{(0)}}{a_{11}} = \frac{1 - 0 - 0}{2} = 1/2$$

$$x_2^{(1)} = \frac{b_2 - a_{21} x_1^{(0)} - a_{23} x_3^{(0)}}{a_{22}} = \frac{8 - 0 - 0}{3} = 8/3$$

$$x_3^{(1)} = \frac{b_3 - a_{31} x_1^{(0)} - a_{32} x_2^{(0)}}{a_{33}} = \frac{-5 - 0 - 0}{2} = -5/2$$

$$\Rightarrow x^{(1)} = \begin{pmatrix} 1/2 \\ 8/3 \\ -5/2 \end{pmatrix}.$$

$$x_1^{(2)} = \frac{b_1 - a_{12} x_2^{(1)} - a_{13} x_3^{(1)}}{a_{11}} = \frac{1 - (-1) \cdot 8/3 - 0 \cdot (-5/2)}{2} = \frac{11}{6}$$

$$x_2^{(2)} = \frac{b_2 - a_{21} x_1^{(1)} - a_{23} x_3^{(1)}}{a_{22}} = \frac{8 - (-1) 1/2 - (-1) \cdot (-5/2)}{3} = 2$$

$$x_3^{(2)} = \frac{b_3 - a_{31} x_1^{(1)} - a_{32} x_2^{(1)}}{a_{33}} = \frac{-5 - 0 \cdot - (-1) 8/3}{2} = -7/6.$$

$$\Rightarrow x^{(2)} = \begin{pmatrix} 11/6 \\ 2 \\ -7/6 \end{pmatrix}.$$

$$x_1^{(3)} = \frac{b_1 - a_{12} x_2^{(2)} - a_{13} x_3^{(2)}}{a_{11}} = \frac{1 - (-1)(2) - 0}{2} = 3/2$$

$$x_2^{(3)} = \frac{b_2 - a_{21} x_1^{(2)} - a_{23} x_3^{(2)}}{a_{22}} = \frac{8 - (-1)(1/6) - (-1)(-7/6)}{3} = \frac{26}{9}$$

$$x_3^{(3)} = \frac{b_3 - a_{31} x_1^{(2)} - a_{32} x_2^{(2)}}{a_{33}} = \frac{-5 - 0 - (-1) 2}{2} = -3/2.$$

$$x^{(3)} = \begin{pmatrix} 3/2 \\ 26/9 \\ -3/2 \end{pmatrix}$$

The method of Gauss-Seidel is similar to Jacobi's except that during the k-th iteration, while updating $x_i^{(k)} \to x_i^{(k+1)}$, we use the updated values $x_1^{(k+1)}, \dots, x_{i-1}^{(k+1)}$ instead of the "old" values $x_1^{(k)}, \dots, x_{i-1}^{(k)}$.

$$a_{11}x_1^{(k+1)} + a_{12}x_2^{(k)} + a_{13}x_3^{(k)} + \cdots + a_{1n}x_n^{(k)} = b_1$$
$$a_{21}x_1^{(k+1)} + a_{22}x_2^{(k+1)} + a_{23}x_3^{(k)} + \cdots + a_{2n}x_n^{(k)} = b_2$$
$$a_{31}x_1^{(k+1)} + a_{32}x_2^{(k+1)} + a_{33}x_3^{(k+1)} + a_{34}x_4^{(k)} + \cdots + a_{3n}x_n^{(k)} = b_3$$
$$\vdots$$
$$a_{i1}x_1^{(k+1)} + \cdots + a_{i,i-1}x_{i-1}^{(k+1)} + a_{ii}x_i^{(k+1)} + a_{i,i+1}x_{i+1}^{(k)} + \cdots + a_{in}x_n^{(k)} = b_i$$
$$\vdots$$
$$a_{n1}x_1^{(k+1)} + \cdots + a_{n,n-1}x_{n-1}^{(k+1)} + a_{nn}x_n^{(k+1)} = b_n$$

$$\Rightarrow x_i^{(k+1)} = \frac{b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}}{a_{ii}}, \quad i = 1, \dots, n.$$

For the previous example,

$$x_1^{(1)} = \frac{b_1 - a_{12}x_2^{(0)} - a_{13}x_3^{(0)}}{a_{11}} = \frac{1-0-0}{2} = \frac{1}{2}$$

$$x_2^{(1)} = \frac{b_2 - a_{21}x_1^{(1)} - a_{23}x_3^{(0)}}{a_{22}} = \frac{8-(-1)(1/2)-0}{3} = \frac{17}{6}$$

$$x_3^{(1)} = \frac{b_3 - a_{31}x_1^{(1)} - a_{32}x_2^{(1)}}{a_{33}} = \frac{-5-0-(-1)\cdot 17/6}{2} = -\frac{13}{12}.$$

General framework for iterative methods

Many iterative methods, including Jacobi and Gauss-Seidel can be described in a unified manner as follows: Select a nonsingular matrix Q. (The choice of Q specifies the method)

Split the system $Ax = b$ as follows by writing $A = Q - (Q - A)$

$$[Q - (Q - A)]x = b \Rightarrow \boxed{Qx = (Q - A)x + b}$$

This splitting motivates an iterative method:

$$\begin{cases} Q x^{(k+1)} = (Q - A) x^{(k)} + b, & k = 0, 1, \ldots \\ x^{(0)} \text{ given} \end{cases}$$

$$\Rightarrow \boxed{x^{(k+1)} = [I - Q^{-1}A] x^{(k)} + Q^{-1} b}$$

$I - Q^{-1}A$ is called the iteration matrix

For Jacobi, $Q = D$ the diagonal part of A

$$A = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 3 & -1 \\ 0 & -1 & 2 \end{bmatrix} \Rightarrow Q = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 2 \end{bmatrix}; \quad Q - A = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$Q^{-1} = \begin{bmatrix} 1/2 & & \\ & 1/3 & \\ & & 1/2 \end{bmatrix} \Rightarrow x^{(k+1)} = [I - Q^{-1}A] x^{(k)} + Q^{-1} b$$

$$= \begin{bmatrix} 0 & 1/2 & 0 \\ 1/3 & 0 & 1/3 \\ 0 & 1/2 & 0 \end{bmatrix} x^{(k)} + \begin{pmatrix} 1/2 \\ 8/3 \\ -5/2 \end{pmatrix}$$

$A \equiv D - L - U$
- D diagonal part of A
- $-L$ strictly lower triangular part of A
- $-U$ " upper " " " A

For Gauss-Seidel Q is the lower triangular part of A

$$\Rightarrow Q = \begin{bmatrix} 2 & 0 & 0 \\ -1 & 3 & 0 \\ 0 & -1 & 2 \end{bmatrix} \Rightarrow Q^{-1} = \begin{bmatrix} \frac{1}{2} & 0 & 0 \\ \frac{1}{6} & \frac{1}{3} & 0 \\ \frac{1}{12} & \frac{1}{6} & \frac{1}{2} \end{bmatrix}$$

$(Q = D - L)$

Inverse of a lower (upper) triangular matrix is also lower (upper) triangular!

$$\Rightarrow \mathcal{G} \equiv I - Q^{-1}A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} \frac{1}{2} & 0 & 0 \\ \frac{1}{6} & \frac{1}{3} & 0 \\ \frac{1}{12} & \frac{1}{6} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} 2 & -1 & 0 \\ -1 & 3 & -1 \\ 0 & -1 & 2 \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{6} & \frac{1}{3} \\ 0 & \frac{1}{12} & \frac{1}{6} \end{bmatrix}$$

$$\Rightarrow x^{(k+1)} = \begin{bmatrix} 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{6} & \frac{1}{3} \\ 0 & \frac{1}{12} & \frac{1}{6} \end{bmatrix} x^{(k)} + \begin{bmatrix} \frac{1}{2} \\ \frac{17}{6} \\ \frac{-13}{12} \end{bmatrix}.$$

Remark For Gauss-Seidel, the first column of $I - Q^{-1}A$ is always zero.

The SOR (successive Overrelaxation Method)

Let $\omega > 0$ be a real number the relaxation parameter

The SOR one step is defined by taking $\boxed{Q = \frac{1}{\omega} D - L}$

$$\Rightarrow \left(\frac{1}{\omega} D - L\right) x^{(k+1)} = \left(\frac{1-\omega}{\omega} D + U\right) x^{(k+1)} + b$$

Remarks (i) If $\omega = 1$ then SOR = Gauss-Seidel.

(ii) For appropriately chosen ω, SOR converges much faster than Jacobi or Gauss-Seidel, i.e. will require far fewer iterations to achieve the same error

For $A = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 3 & -1 \\ 0 & -1 & 2 \end{bmatrix}$, $Q = \frac{1}{\omega} D - L = \begin{bmatrix} 2/\omega & 0 & 0 \\ -1 & 3/\omega & 0 \\ 0 & -1 & 2/\omega \end{bmatrix}$

$$Q - A = \begin{bmatrix} \frac{2(1-\omega)}{\omega} & 1 & 0 \\ 0 & \frac{3(1-\omega)}{\omega} & 1 \\ 0 & 0 & \frac{2(1-\omega)}{\omega} \end{bmatrix}.$$

Ex.

$$A = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 3 & -1 \\ 0 & -1 & 2 \end{bmatrix}; \quad b = \begin{bmatrix} 1 \\ 8 \\ -5 \end{bmatrix}, \quad x = \begin{bmatrix} 2 \\ 3 \\ -1 \end{bmatrix}$$

The columns below show the errors $\|x^{(k)} - x\|$ for Jacobi, Gauss-Seidel and SOR with $\omega = \frac{2\sqrt{3}}{\sqrt{3}+\sqrt{2}}$. Here

$$\|v\| = \max_{1 \le i \le n} |v_i| \quad \Rightarrow \quad \|x^{(k)} - x\| = \max_{1 \le i \le n} |x_i^{(k)} - x_i|.$$

	Jacobi	Gauss-Seidel	SOR
k = 0	0.3000000D+01	0.3000000D+01	0.3000000D+01
k = 1	0.1500000D+01	0.1500000D+01	0.1449490D+01
k = 2	0.1000000D+01	0.8333333D-01	0.2224513D+00
k = 3	0.5000000D+00	0.2777778D-01	0.1041343D-01
k = 4	0.3333333D+00	0.9259259D-02	0.4748586D-02
k = 5	0.1666667D+00	0.3086420D-02	0.3497890D-03
k = 6	0.1111111D+00	0.1028807D-02	0.7375282D-04
k = 7	0.5555556D-01	0.3429355D-03	0.6124741D-05
k = 8	0.3703704D-01	0.1143118D-03	0.1010775D-05
k = 9	0.1851852D-01	0.3810395D-04	0.8857896D-07
k = 10	0.1234568D-01	0.1270132D-04	0.1294922D-07
k = 11	0.6172840D-02	0.4233772D-05	0.1170061D-08
k = 12	0.4115226D-02	0.1411257D-05	0.1590306D-09
k = 13	0.2057613D-02	0.4704191D-06	0.1465628D-10
k = 14	0.1371742D-02	0.1568064D-06	0.1897149D-11
k = 15	0.6858711D-03	0.5226879D-07	0.1771916D-12
k = 16	0.4572474D-03	0.1742293D-07	0.2220446D-13
k = 17	0.2286237D-03	0.5807643D-08	0.2664535D-14
k = 18	0.1524158D-03	0.1935881D-08	0.4440892D-15
k = 19	0.7620790D-04	0.6452936D-09	0.0000000D+00
k = 20	0.5080526D-04	0.2150979D-09	0.0000000D+00

At each iteration,

The error for Jacobi is reduced by a factor of .66

" " " Gauss-Seidel " " " " " " " .33 ---

" " " SOR " " " " " " " .12

Defn. Given a square matrix M, a scalar λ (could be real or complex) is called an eigenvalue of M if for some nonzero vector v, $Mv = \lambda v$. The eigenvector corresponding to λ, such that $Mv = \lambda v$.

The eigenvalues of an $n \times n$ matrix M are roots of its characteristic polynomial $p(\lambda) = \det(\lambda I - M)$. $p(\lambda)$ has degree n thus has exactly n roots, i.e. eigenvalues. These eigenvalues may not be all distinct, so we have to count multiplicities.

The collection of all eigenvalues of a matrix M is called the spectrum of M, $\sigma(M)$. The largest eigenvalue of M in modulus (recall they may be complex) is called the spectral radius of M:

$$\rho = \max_{\lambda \in \sigma(M)} |\lambda|$$

Spectral radius Theorem For the sequence $\{x^{(k)}\}_{k \geq 0}$ obtained by the general iterative method $x^{(k+1)} = (I - Q^{-1}A)x^{(k)} + Q^{-1}b$ to converge, no matter what starting vector $x^{(0)}$ is selected, it is necessary and sufficient that $\rho(I - Q^{-1}A) < 1$. □

Remark The smaller $\rho(I - Q^{-1}A)$ the faster the convergence. Indeed, it can be shown that the errors $\|x^{(k)} - x\| = \max_{1 \le i \le n} |x_i^{(k)} - x_i|$ decrease, on the average, as ρ^k.

For Jacobi $\rho(I - Q^{-1}A) = \frac{1}{\sqrt{3}} \approx .5774$

For Gauss-Seidel $\rho(I - Q^{-1}A) = \frac{1}{3} \approx .33 = \frac{1}{10}$.

For SOR the optimal parameter $\omega = 1.101 + 0.205$

and $\rho(I - Q^{-1}A) = \omega - 1 = .101$

This means that SOR with the optimal parameter will require 3 times less iterations than Gauss-Seidel.

Remark These figures are valid only for the matrix in the above ex.

Theorem If A is strictly diagonally dominant, then the Jacobi and Gauss-Seidel methods converge for any $x^{(0)}$.

Theorem Suppose A is symmetric positive definite. Then the SOR method converges for any $x^{(0)}$ if $0 < \omega < 2$.
In particular, the Gauss-Seidel method ($\omega = 1$) is convergent.