2.10 Sistemas de ecuaciones lineales y combinaciones lineales de vectores

Otra forma de representar el sistema de ecuaciones lineales (2)

$$x_1\begin{pmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{m1} \end{pmatrix} + x_2\begin{pmatrix} a_{12} \\ a_{22} \\ \vdots \\ a_{m2} \end{pmatrix} + \ldots + x_n\begin{pmatrix} a_{1n} \\ a_{2n} \\ \vdots \\ a_{mn} \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}$$

Sea un vector $v \in k^m$ es una combinacion lineal de vectores $u_1, u_2, ..., u_n$ en k^m, con escalares $\alpha_1, \alpha_2, ..., \alpha_n$ tal que

$$\alpha_1 u_1 + \alpha_2 u_2 + ... + \alpha_n u_n = v$$

2.10 Sistemas de ecuaciones lineales y combinaciones lineales de vectores

Otra forma de representar el sistema de ecuaciones lineales (2)

$$x_1\begin{pmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{m1} \end{pmatrix} + x_2\begin{pmatrix} a_{12} \\ a_{22} \\ \vdots \\ a_{m2} \end{pmatrix} + \ldots + x_n\begin{pmatrix} a_{1n} \\ a_{2n} \\ \vdots \\ a_{mn} \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}$$

$$\alpha_1 \; u_1 \; + \; \alpha_2 \; u_2 \; + \; \ldots + \alpha_n \; u_n \; = \; v$$

Entonces el sistema de ecuaciones lineales (2) y la ecuación vectorial equivalente tienen solución si el vector columna de constantes (i.e.*v*.) es una combinación lineal de las columnas de la matriz de coeficientes (i.e. familia de u's)

<u>Teorema</u>

Un sistema *Ax*=*B* de ecuaciones lineales tiene solución si y solo si *B* es una combinación lineal de las columnas de la matriz de coeficientes A.

Ejemplo:

Escribir $v = (1, -2, 5)$ como una combinacion de vectores

$u_1 = (1,1,1), u_2 = (1,2,3), \ u_3 = (2,-1,1),$

i.e. $v = xu_1 + yu_2 + zu_3$

x, y, z incognitas que buscamos

Desarrollando,

$$\begin{pmatrix} 1 \\ -2 \\ 5 \end{pmatrix} = x\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} + y\begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} + z\begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix} \longrightarrow \begin{pmatrix} 1 \\ -2 \\ 5 \end{pmatrix} = \begin{pmatrix} x + y + 2z \\ x + 2y - z \\ x + 3y + z \end{pmatrix}$$

resolviendo primero a forma escalonada

$$\begin{aligned} x + y + 2z &= 1 \\ y - 3z &= -3 \\ 2y - z &= 4 \end{aligned}$$

$$\begin{aligned} x + y + 2z &= 1 \\ y - 3z &= -3 \\ 5z &= 10 \end{aligned}$$

Segundo: sustitución hacia atrás

$Z=2$ $y=3$ $x=-6$

Ejemplo (representación de vectores como combinación lineal de un conjunto de vectores)

Considerar
$$\left.\begin{aligned} u_1 &= (1,\ 1,\ 1) \\ u_2 &= (1, -3,\ 2) \\ u_3 &= (5, -1, -4) \end{aligned}\right\} \text{vectores en } \mathbb{R}^3$$

u_1, u_2, u_3 son ortogonales i.e
$$u_1 \cdot u_2 = 0$$
$$u_1 \cdot u_3 = 0$$
$$u_2 \cdot u_3 = 0$$

Queremos escribir $v = (4,\ 14, -9)$ como combinacion lineal de u_1, u_2 y u_3

•Método 1. Encontrar el sistema de ecuaciones lineales y resolver

$$\begin{aligned} x + y + 5z &= 4 \\ x - 3y - z &= 14 \\ x + 2y - 4z &= -9 \end{aligned}$$

1ero Eliminacion hacia adelante

$$\begin{aligned} x + y + 5z &= 4 \\ -4y - 6z &= 10 \\ y - 9z &= -13 \end{aligned}$$

$$\begin{aligned} x + y + 5z &= 4 \\ y - 9z &= -13 \\ -4y - 6z &= 10 \end{aligned}$$

$$\begin{aligned} x + y + 5z &= 4 \\ y - 9z &= -13 \\ -42z &= -42 \end{aligned}$$

2^{do} Sustitucion hacia atras

$$\begin{aligned} z &= 1 \\ y &= -4 \\ x &= 3 \end{aligned}$$

$$\therefore \quad v = 3u_1 - 4u_2 + u_3$$

Método 2. (explotando la ortogonalidad de los vectores $\rightarrow$ aritmética más sencilla)

Queremos resolver

$$(4,14,-9) = x(1,1,1) + y(1,-3,2) + z(5,-1,-4) \quad \text{(a)}$$

Producto punto de (a) con u_1

$$(4,14,-9)\cdot(1,1,1) = x(1,1,1)\cdot(1,1,1)$$

$$4+14-9 = x(1+1+1)$$

$$9 = 3x$$

$$x = 3$$

Los otros dos sumandos son cero porque los vectores son ortogonales

Mismo para los demás productos punto. Resultado *y=-4* , *z=1*

Podemos generalizar para $\mathbb{R}^n$ esta forma de representar la combinación lineal de Vectores cuando sabemos que son mutuamente ortogonales

Teorema.

Sean $u_1, u_2, \ldots, u_n$ vectores mutuamente ortogonales diferentes de cero en $\mathbb{R}^{\mathrm{n}}$ (i.e. $u_i \cdot u_i \neq 0$)

Entonces para cualquier vector $v \in \mathbb{R}^n$

$$v = \frac{v \cdot u_1}{u_1 \cdot u_1} u_1 + \frac{v \cdot u_2}{u_2 \cdot u_2} u_2 + \ldots + \frac{v \cdot u_n}{u_n \cdot u_n} u_n$$

Prueba

Suponer v como combinacion lineal de vectores $u_1, u_2, \ldots, u_n \in \mathbb{R}^{\mathrm{n}}$

Entonces $v = a_1 u_1 + a_2 u_2 + \ldots + a_n u_n \quad (*)$

donde $a_1, a_2, \ldots, a_n \in \mathbb{R}$ o $\mathbb{C}$

$$v \cdot u_1 = a_1 u_1 \cdot u_1 + a_2 u_2 \cdot u_1 + \ldots + a_n u_n \cdot u_1$$

$$v \cdot u_1 = a_1 u_1 \cdot u_1 + \cancelto{0}{a_2} u_2 \cdot u_1 + \ldots + \cancelto{0}{a_n} u_n \cdot u_1$$

$$a_1 = \frac{v\cdot u_1}{u_1\cdot u_1}$$

$$v\cdot u_2 = a_1u_1\cdot u_2 + a_2u_2\cdot u_2 + \ldots + a_nu_n\cdot u_2$$

$$v\cdot u_2 = a_1u_1\cdot u_2 + \cancelto{0}{a_2}u_2\cdot u_2 + \ldots + \cancelto{0}{a_n}u_n\cdot u_2$$

$$a_2 = \frac{v\cdot u_2}{u_2\cdot u_2}$$

repetir hasta n y sustituir en (*)

$$v = \frac{v\cdot u_1}{\underbrace{u_1\cdot u_1}}u_1 + \frac{v\cdot u_2}{u_2\cdot u_2}u_2 + \ldots + \frac{v\cdot u_n}{u_n\cdot u_n}u_n$$

Coeficiente de Fourier
de v con respecto a u_i

2.11.- Sistemas Homogéneos y no Homogéneos de Ecuaciones Lineales

- Un sistema de ecuaciones lineales es homogéneo si y solo si el vector columna de constantes es cero. Esto es *Ax=0*
- Sistema siempre con solución trivial (vector cero)
- En que condiciones tiene *Ax=0* otras soluciones?

Sistema homogeneo en forma escalonada

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &= 0 \\ a_{2j_2}x_{j_2} + a_{2n}x_n &= 0 \\ &\vdots \\ a_{rj_r}x_{j_r} + \ldots + a_{rn}x_n &= 0 \end{aligned}$$

r , numero de ecuaciones

n , numero de incognitas

Si $r = n$ solo solución trivial (sistema triangular)

Si $r < n$ solución diferente de cero (debido a variables libres)

No se da el caso de $r > n$ en forma escalonada.

De hecho, si comenzamos con mas incógnitas que ecuaciones, entonces en forma escalonada, $r < n$ se mantiene.
Esto se muestra en el siguiente teorema:

<u>Teorema</u>
El sistema homogéneo $Ax=0$ con más incógnitas que ecuaciones tiene al menos una solución diferente de cero.

Base de la solución general de un sistema homogéneo

Sea W la solucion general de un sistema homogeneo $Ax = 0$.

$W = \{ w \in W | w$ es una combinacion lineal de vectores base$\}$

Una base para W es un conjunto de vectores solucion $u_1, u_2, ..., u_s$ si cada vector solucion $w \in W$ se puede expresar como una combinacion lineal de los vectores $u_1, u_2, ..., u_s$. Esto es, existen escalares $a_1, a_2, ..., a_s$ tal que $w = a_1 u_1 + a_2 u_2 + ... + a_s u_s$

s, número de vectores de la base es igual al número de variables libres.

s, dimensión de W dim(W)=s

Si $w = \{0\}$, solucion cero, se define $\dim W = 0$

¿cómo encontrar esa base?

Siguiente teorema nos lo dice

Teorema

Sea W la solución general del sistema homogéneo $Ax=0$.

Suponer que la forma escalonada del sistema $Ax=0$ tiene s variables libres

Sea $u_1, u_2, ..., u_s$ soluciones haciendo la variable libre i igual a una constante (i.e. a 1) y el resto a cero.

Entonces $\dim W = s$ y la base es $u_1, u_2, ..., u_s$

Ojo: la solución general puede tener muchas bases!!!

Este teorema nos da solo una base.

Sistemas no homogéneos y sistema homogéneo asociado

Sea $Ax=B$ sistema no homogéneo

$Ax=0$ sistema homogéneo asociado

Teorema

La solucion general de $Ax = B$ esta dada por $u = v_0 + w = \{v_0 + w \mid w \in W\}$ donde v_0 es una solucion particular y W es la solucion general del sistema homogeneo. En otras palabras añadimos v_0 a cada elemento de W.

Ejemplos

- Precios de equilibrio en una economía
- Distribuciones de Población

2.12 Complejidad de Algoritmos Gauss y Gauss-Jordan

Método Gauss y Gauss-Jordan O.K. para resolver sistemas de ecuaciones lineales usando lápiz y papel.

En la práctica, sistemas de miles de ecuaciones que se resuelven muchísimas veces

$\rightarrow$ Resolver con computadora

$\rightarrow$con algoritmos eficientes

$\rightarrow$consumo de tiempo

- operaciones aritméticas
- indexación
- manejo de estructuras de datos
- almacenamiento y recuperación

$\rightarrow$ consumo de memoria

Consideraciones de tiempo

Resolver un sistema de ecuaciones lineales $\equiv$ hacer aritmetica

e.g. reduccion de una matriz con operaciones sobre renglones

paso típico $\rightarrow$ reemplazar R_i por $rR_k + R_i$

en pseudo-código:

Hacer con $j = 1$ hasta n

$$a_{ij} = a_{ij} + ra_{kj}$$

esta instrucción incluye operaciones

- adición
- multiplicación
- indexado de variables
- almacenamiento y recuperación de valores

Esto es un flop (c.f. C.B. Moles)

Pregunta: Cuanto tiempo se lleva una PC en ejecutar una

- suma, resta, multiplicación, división, flop

Por ejemplo,

FOR i=1 to N: C=R+S : NEXT

con R, S números reales al azar

+ cambiarlo por otros símbolos: -, ·, /

FOR i=1 to N: *a(k, j) = a(k, j) + R·a(l, j):* NEXT

Lenguaje Basic (1987)

seg. / N= 10, 000 operaciones

	Interpretado		Compilado	
Rutina	Sencillo	Doble	Sencillo	Doble
+	37	44	8	9.00
-	37	48	8	9
*	39	53	9	11
/	47	223	9	15
flop	143	162	15	18

MatLab 6, 2004

seg. / N= 10´000, 000 operaciones

	Interpretado	Compilado en C
Rutina	Doble	Doble
+	15.8	1.30
-	15.4	1.4
*	15.4	2
/	15.6	1.9
flop	43.9	5.7

Comentarios:

1. Multiplicación toma más tiempo que suma (pero no mucho más)

2. División es la que consume más tiempo de las 4 operaciones aritméticas (sobre todo en Basic interpretado)

$\therefore$ tratar de evitar divisiones

por ejemplo si se va a calcular

(4/3) (3, 2, 5, 2, 8) + (-4, 11, 2, 1, 5)

para eliminar primer elemento, evitar calcular el resto de los elementos realizando división 4/3

$$\left.\begin{array}{l}(4/3)\;2+11 \quad ; \\ (4/3)\;5+2 \quad ,\text{ etc.}\end{array}\right\}\text{4 divisiones}$$

preferible hacer $r = 4/3$ y luego $2r+11,\ 5r+2,$ etc

3. Mayor velocidad en lenguajes compilados
(notar diferencia sustancial en doble precisión)

4. Indexado en flops requiere mucho tiempo

Conteo de Operaciones en el Método de Gauss (orden de magnitud de complejidad computacional)

Conteo de flops en la solución del sistema de ecuaciones lineales $Ax=B$ con una única solución, para tener una medida de la complejidad de método de solución.

Sea $Ax=B$ (cuadrado)
Resolver utilizando Gauss con sustitución hacia atrás

Matriz aumentada

$$M = (A \mid B) = \left[\begin{array}{cccc|c} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ \vdots & & & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} & b_n \end{array}\right]$$

Por el momento, ignorar los flops realizados en el vector de constantes B
Contar solo los de A.

1.- Reducción a forma triangular *U*

- Cálculo del pivote en R_1
  - Añadir múltiplo de R_1 a R_2 en $A \rightarrow n$-1 flops
    (no tenemos que calcular el primero, sabemos que es cero)
  - Añadir múltiplo de R1 a R3 en $A \rightarrow n$-1 flops
  - Así sucesivamente para n-1 ecuaciones
  - Por lo tanto, # de flops para encontrar el pivote en R_1 es $(n-1)^2$
- Cálculo del pivote en R_2 con matriz (n-1)x(n-1)
  - Idem cálculo R_1
  - Por lo tanto, # de flops para encontrar el pivote en R_2 es $(n-2)^2$

$\therefore$ Número de flops para reducir a forma triangular U

$(n\text{-}1)^2 + (n\text{-}2)^2 + \dots + 1$ flops

Encontrando una forma compacta para esta expresion...

Sabemos que $1^2 + 2^2 + 3^2 + \ldots + n^2 = \dfrac{n(n+1)(2n+1)}{6}$

sustituyendo n por n -1

$$\frac{(n-1)(n)(2n-1)}{6} = \frac{(n^2-n)(2n-1)}{6} = \underbrace{\frac{n^3}{3} - \frac{n^2}{2} + \frac{n}{6}}_{*}$$

Principal contribución: $\dfrac{n^3}{3}$

Por ejemplo si $n = 1{,}000$ entonces $(*) = \dfrac{1{,}000{,}000{,}000}{3} - \underbrace{\dfrac{1{,}000{,}000}{2}}_{} + \underbrace{\dfrac{1000}{6}}_{}$

estos contribuyen poco a $n^3/3$

Se acostumbra usar $n^3/3$ como medida del orden de magnitud para valores grandes de n.

Ahora, divisiones realizadas para encontrar U.

Si tenemos un buen algoritmo, entonces 1 división por cada operación de renglón (como máximo)

$\therefore$ $(n-1)$ divisiones para encontrar pivote 1

$(n-2)$ divisiones para encontrar pivote 2, etc.

Total de divisiones $(n\text{-}1)+(n\text{-}2)+(n\text{-}3)+\ldots+1$

Encontrando una expresión compacta para esto sabemos que

$$1+2+3+\ldots+n=\frac{n(n+1)}{2}$$

Sustituyendo n por n-1

$$\frac{(n-1)n}{2}=\frac{n^2-n}{2}=\frac{n^2}{2}-\frac{n}{2}$$

$\therefore$ Para valores grandes de n, el orden de magnitud $n^2/2$ no se compara con $n^3/3$ de los flops.

Orden de magnitud para las operaciones de B es $n^2/2$

$\therefore$ No impacta en el total de operaciones

Resumiendo...

<u>Cuenta de flops para reducir $(A|B)$ a $(U|c)$</u>

Si $Ax=B$ es un sistema de ecuaciones lineales cuadrado con A orden n, entonces el número de flops ejecutado para reducir $(A|b)$ a $(U|c)$ es del orden de magnitud $n^3/3$ para n grandes.

Cálculo del número de flops para sustitución hacia atrás resolviendo $Ux=c$

Tenemos el siguiente sistema

$$
\begin{aligned}
u_{11}x_1 + \ldots + u_{1,n-1}x_{n-1} + u_{1,n}x_n &= c_1 \\
&\vdots \\
u_{n-1,n-1}x_{n-1} + u_{n-1,n}x_n &= c_{n-1} \\
u_{n,n}x_n &= c_n
\end{aligned}
$$

Resolviendo para x_n $\rightarrow$ una división indexada $\equiv$ un flop

Resolviendo para x_{n-1} $\rightarrow$ una multiplicación (sust. valor x_n)
una suma (añadir c_{n-1})
ambas indexadas
$\equiv$ un flop

una división indexada (obtener x_{n-1}) $\equiv$ un flop

2 flops

para $x_{n-2} \rightarrow$ 2 flops multiplicación y suma

1 flop división

3 flops

Por lo tanto, obtenemos

$$1+2+3+\ldots+n=\frac{(n+1)n}{2}=\frac{n^2}{2}+\frac{n}{2}$$

De nuevo esto es de menor orden de magnitud que $n^3/3$ obtenido en la reducción de $(A|B)$ a $(U|c)$.

$\therefore$ Resolver $Ax=B$ (A cuadrada de orden n) utilizando Gauss con substitución hacia atrás se logra en un orden de magnitud de $n^3/3$ para n grandes.

Complejidad del algoritmo Gauss-Jordan

Gauss-Jordan $\rightarrow$ reduce $Ax=B$ a $Ix=C$

Realizando un proceso similar al arriba elaborado encontramos orden $n^3/2$.

Por lo tanto, para n grandes Gauss es más eficiente

Si n es proporcional al tiempo, entonces

$$tg = \frac{n^3}{3}; tgj = \frac{n^3}{2}$$

$$3tg = n^3 = 2tgj$$

$$tgj = 1.5tg$$

Algoritmo Gauss-Jordan = 1.5 (Algoritmo Gauss)

2.13.- Factorización LU y complejidad

(o hacer un registro de las operaciones elementales por renglones)

- Continuamos trabajando con sistema cuadrado $Ax=b$ con solución única. Solución se encuentra con Gauss sin necesidad de cambiar renglones. (situación común en la práctica)

- En ocasiones, necesario resolver muchas veces un mismo sistema con diferente vector de constantes
  $\rightarrow$ resolver $(A|b_1, b_2, b_3, ..., b_s)$
  problemas:
  - s muy grande $\rightarrow$ no espacio en la memoria
  - b_i generado en un periodo de tiempo
    $\rightarrow$ resolver por lotes cada periodo de tiempo
  - b_{i+1} depende de la solución de $Ax=b_i$
    $\rightarrow$ necesario resolver secuencialmente

De sección anterior sabemos que

- magnitud de número de flops para reducir una matriz *A* a su forma triangular *U* es $n^3/3$ para n grandes.

$\therefore$ queremos evitar repetir este trabajo cada vez

Suponemos que *A* puede ser reducida a *U* sin intercambiar renglones (esto es, pivotes aparecen donde nosotros queremos cuando se reduce de *A* a *U*).

$\therefore$ puras operaciones sobre renglones

Esto es, añadir un múltiplo de un renglón a otro renglón.

Esto es equivalente a multiplicar por la izquierda a *A* por matriz *E* de orden *n*

E se obtiene aplicando la misma operación de renglón a la matriz *I*.

Ejemplo

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & -5 & 6 \\ 7 & 8 & 9 \end{pmatrix} \sim \begin{pmatrix} 1 & 2 & 3 \\ 0 & -13 & -6 \\ 7 & 8 & 9 \end{pmatrix}$$

$$L_2 \leftarrow -4L_1 + L_2$$

$$I = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 0 & 0 \\ -4 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = E_1$$

$$E_1 A = \begin{pmatrix} 1 & 0 & 0 \\ -4 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 \\ 4 & -5 & 6 \\ 7 & 8 & 9 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 \\ 4-4 & -8-5 & 6-12 \\ 7 & 8 & 9 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 \\ 0 & -13 & -6 \\ 7 & 8 & 9 \end{pmatrix}$$

Extendiendo esta idea...

$E_n\ E_{n-1} \ldots E_2\ E_1\ A = U$

Si estamos resolviendo para diferentes b_i podríamos aplicar la secuencia $E_n\ E_{n-1} \ldots E_2\ E_1$ a cada b_i.

¿Cómo mantener un registro de estas operaciones de manera eficiente?

Respuesta: creando una matriz -L

Procedimiento

- Iniciar con I orden n
- Si durante la reducción de A a U, renglón Ri es multiplicado por r y se añade a renglón Rk, reemplazar el cero de renglón *k* y columna *i* de I por – r.
- El resultado final es matriz L

Ejemplo

$$A=\begin{pmatrix}1&-2&0&3\\-2&3&1&-6\\-1&4&-4&3\\5&-8&4&0\end{pmatrix}\qquad b=\begin{pmatrix}11\\-21\\-1\\23\end{pmatrix}$$

$$A=\begin{pmatrix}1&-2&0&3\\-2&3&1&-6\\-1&4&-4&3\\5&-8&4&0\end{pmatrix}\qquad I=\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{pmatrix}$$

$$\sim\begin{pmatrix}1&-2&0&3\\0&-1&1&0\\0&2&-4&6\\0&2&4&-15\end{pmatrix}\qquad\sim\begin{pmatrix}1&0&0&0\\2&1&0&0\\1&0&1&0\\-5&0&0&1\end{pmatrix}$$

$$\sim\begin{pmatrix}1&-2&0&3\\0&-1&1&0\\0&0&-2&6\\0&0&6&-15\end{pmatrix}\qquad\sim\begin{pmatrix}1&0&0&0\\2&1&0&0\\1&2&1&0\\-5&2&0&1\end{pmatrix}$$

$$\sim\begin{pmatrix}1&-2&0&3\\0&-1&1&0\\0&0&-2&6\\0&0&0&3\end{pmatrix}\qquad\sim\begin{pmatrix}1&0&0&0\\2&1&0&0\\1&2&1&0\\-5&2&3&1\end{pmatrix}=-L$$

$$L=\begin{pmatrix}1&0&0&0\\-2&1&0&0\\-1&-2&1&0\\5&-2&-3&1\end{pmatrix}$$

Aplicar "registro" –L en vector b. Esto es, aplicar las operaciones indicadas en cada columna para cada renglón

$$-L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 1 & 2 & 1 & 0 \\ -5 & 2 & 3 & 1 \end{pmatrix}$$

1a col

$$b = \begin{pmatrix} 11 \\ -21 \\ -1 \\ 23 \end{pmatrix} \sim \overset{\text{x2 en } R_2}{\begin{pmatrix} 11 \\ 1 \\ -1 \\ 23 \end{pmatrix}} \sim \overset{\text{x1 en } R_3}{\begin{pmatrix} 11 \\ 1 \\ 10 \\ 23 \end{pmatrix}} \sim \overset{\text{x(-5) en } R_4}{\begin{pmatrix} 11 \\ 1 \\ 10 \\ -32 \end{pmatrix}}$$

2a col

$$\sim \begin{pmatrix} 11 \\ 1 \\ 10 \\ -32 \end{pmatrix} \sim \overset{\text{x2 en } R_3}{\begin{pmatrix} 11 \\ 1 \\ 12 \\ -32 \end{pmatrix}} \sim \overset{\text{x2 en } R_4}{\begin{pmatrix} 11 \\ 1 \\ 12 \\ -30 \end{pmatrix}}$$

3a col

$$\sim \begin{pmatrix} 11 \\ 1 \\ 12 \\ -30 \end{pmatrix} \sim \overset{\text{x3 en } R_4}{\begin{pmatrix} 11 \\ 1 \\ 12 \\ 6 \end{pmatrix}}$$

$$\therefore \left[U \middle| c \right] = \left(\begin{array}{cccc|c} 1 & -2 & 0 & 3 & 11 \\ 0 & -1 & 1 & 0 & 1 \\ 0 & 0 & -2 & 6 & 12 \\ 0 & 0 & 0 & 3 & 6 \end{array} \right)$$

Preguntas

1. Para que ponemos 1's en la diagonal si no los usamos?
2. Si multiplicamos R_i en A por r, porqué ponemos - r en L y luego usamos r para resolver b?

Respuestas a continuación

1.- Resolviendo en computadora no usamos diagonal de L.

De hecho, no se crea una matriz L separada .

Cuando se trabaja con arreglos, se utiliza A.

e.j. anterior

$$A = \begin{pmatrix} 1 & -2 & 0 & 3 \\ -2 & 3 & 1 & -6 \\ -1 & 4 & -4 & 3 \\ 5 & -8 & 4 & 0 \end{pmatrix}$$

Representación combinada L\U

$$\begin{pmatrix} 1 & -2 & 0 & 3 \\ -2 & -1 & 1 & 0 \\ -1 & -2 & -2 & 6 \\ 5 & -2 & -3 & 3 \end{pmatrix}$$

Ojo, algunos usan -L

Eficiencia de solución de $Ax = b$ si L y U son conocidas

Para encontrar L y
Para encontrar solución de $Ax = b$ (i.e. operar sobre b)
se realiza un flop por cada elemento de L

Para columna 1 --- n-1 flops
2 --- n-2 flops
:
n–1 --- 1 flop

$\therefore$ número total de flops = $(n-1)+(n-2)+\ldots+2+1=\dfrac{(n-1)n}{2}=\dfrac{n^2}{2}-\dfrac{n}{2}$

mismo orden de magnitud $n^2/2$ para n grandes en sustitución hacia atrás.

Entonces para resolver Ax=b se requieren orden de magnitud $o(n^2/2)$.

Ya que encontrar U es de orden $n^3/3$ con Gauss, entonces usar L\U representación es más eficiente un orden de magnitud menor.

(sin considerar cuestiones de manejo de memoria)

Otras formas de resolver sistema de orden n.

- Cálculo de $x = A^{-1}b$ es de orden n^2 pero cálculo de A^{-1} es de orden $O(n^3)$!!

Otro ejemplo del uso de LU

- Resolver $A^3 x = b$

$$A = \begin{pmatrix} 1 & 2 & -1 \\ -2 & -5 & 3 \\ -1 & -3 & 0 \end{pmatrix}; b = \begin{pmatrix} 9 \\ -17 \\ -44 \end{pmatrix}$$

- Estrategia:

$$A^3 x = b$$

si $y = Ax$, entonces $A^2 y = b$

si $z = Ay$, entonces $Az = b$

Por lo tanto, utilizar fact. LU

Resolver para $Az=b$

Luego, resolver $z=Ay$

Luego, resolver $y=Ax$

Otro ejemplo del uso de LU

- Resolver $A^3 x = b$

$$A = \begin{pmatrix} 1 & 2 & -1 \\ -2 & -5 & 3 \\ -1 & -3 & 0 \end{pmatrix}; b = \begin{pmatrix} 9 \\ -17 \\ -44 \end{pmatrix}$$

$$U = \begin{pmatrix} 1 & 2 & -1 \\ 0 & -1 & 1 \\ 0 & 0 & -2 \end{pmatrix}; \quad -L = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & -1 & 1 \end{pmatrix}$$

Aplicando el registro –L en b

$$b = \begin{pmatrix} 9 \\ -17 \\ -44 \end{pmatrix} \sim \begin{pmatrix} 9 \\ 1 \\ -35 \end{pmatrix} \sim \begin{pmatrix} 9 \\ 1 \\ -36 \end{pmatrix}$$

Resolviendo hacia atrás con U

$$z = \begin{pmatrix} -7 \\ 17 \\ 18 \end{pmatrix}$$

Ahora, resolver $Ay=z$, aplicando el registro L en z

$$z = \begin{pmatrix} -7 \\ 17 \\ 18 \end{pmatrix} \sim \begin{pmatrix} -7 \\ 3 \\ 11 \end{pmatrix} \sim \begin{pmatrix} -7 \\ 3 \\ 8 \end{pmatrix}$$

Resolviendo hacia atrás con U

$$y = \begin{pmatrix} 3 \\ -7 \\ -4 \end{pmatrix}$$

Ahora, resolver $Ax=y$, aplicando el registro L en y

$$y = \begin{pmatrix} 3 \\ -7 \\ -4 \end{pmatrix} \sim \begin{pmatrix} 3 \\ -1 \\ -1 \end{pmatrix} \sim \begin{pmatrix} 3 \\ -1 \\ 0 \end{pmatrix}$$

Resolviendo hacia atrás con U

$$x = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$$

Factorización LU

Sabemos que A=LU

$$A = \begin{pmatrix} 1 & -2 & 0 & 3 \\ -2 & 3 & 1 & -6 \\ -1 & 4 & -4 & 3 \\ 5 & -8 & 4 & 0 \end{pmatrix} \qquad LU = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -2 & 1 & 0 & 0 \\ -1 & -2 & 1 & 0 \\ 5 & -2 & -3 & 1 \end{pmatrix} \begin{pmatrix} 1 & -2 & 0 & 3 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -2 & 6 \\ 0 & 0 & 0 & 3 \end{pmatrix}$$

Si $E_n E_{n-1} \dots E_2 E_1 A = U$

Entonces $A = \underline{E_1^{-1} E_2^{-1} \dots E_{n-1}^{-1} E_n^{-1} U}$

Entonces, esto debe ser L

Sabemos que E_i se obtiene de matriz I

$$\text{operaciones elementales}\begin{cases}\text{- multiplicar un renglon a por } r \\ \text{- sumar resultado a otro renglon } b\end{cases}$$

$$L_b \leftarrow rL_a + L_b$$

$$R_2 \leftarrow -4R_1 + R_2$$

$$I = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \rightarrow E = \begin{pmatrix} 1 & 0 & 0 \\ 4 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$E^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ -4 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$E\ E^{-1} = I$$

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

Sabemos que E_i se obtiene de matriz I

$$\text{operaciones elementales}\begin{cases}\text{- multiplicar un renglon a por } r\\ \text{- sumar resultado a otro renglon } b\end{cases}$$

$$L_b \leftarrow rL_a + L_b$$

Entonces, inversa E_i^{-1} se obtiene de matriz I

- multiplicar el renglón a por –r
- sumar resultado a otro renglón b

$$i.e.\quad L_b \leftarrow -rL_a + L_b$$

En otras palabras, si multiplico renglón R_a por r y lo sumo a R_b para obtener E_i^{-1} pongo el negativo del multiplicador en el renglón b columna a

Algunas cuestiones

- ¿Se puede obtener el factor LU de una matriz cuadrada que no tiene inversa?
- Factorización LU de una matriz cuadrada A invertible no es única.
- ¿Cómo hacerle para encontrar factorización única?

Sea D matriz cuadrada con la misma diagonal que U

Si se obtiene U=DU* donde U* tiene en su diagonal solo 1’s, entonces A=LDU* factorización única

Teorema Factorización única

Sea A matriz cuadrada de orden n

A=LDU es única donde

L es matriz diagonal inferior con diagonal principal en 1's

U es matriz diagonal superior con diagonal principal en 1's

D es matriz con diagonal principal diferente de ceros

Prueba

Suponer $A=L_1D_1U_1$ y $A=L_2D_2U_2$

Entonces mostrar que $L_1=L_2$, $D_1=D_2$ y $U_1=U_2$

L_1^{-1} y L_2^{-1} son también matrices triangulares inferiores (porque operaciones sobre L_1 y L_2 solo afectan a los elementos debajo de la diagonal)

$$\left.\begin{array}{l} D_1^{-1} \text{ y } D_2^{-1} \quad \text{son tambien matrices diagonales} \\ U_1^{-1} \text{ y } \mathrm{U}_2^{-1} \quad \text{son tambien matrices diagonales} \end{array}\right\} \text{por la misma razon ...}$$

... esto se puede ver al pensar como se hace la reducción para encontrar la inversa.

Como $A = L_1 D_1 U_1$ y $A = L_2 D_2 U_2$

Entonces $L_1 D_1 U_1 = L_2 D_2 U_2$

$$L_2^{-1} L_1 D_1 U_1 = L_2^{-1} L_2 D_2 U_2$$

$$L_2^{-1} L_1 D_1 U_1 = D_2 U_2$$

$$L_2^{-1} L_1 D_1 U_1 U_1^{-1} = D_2 U_2 U_1^{-1}$$

$$L_2^{-1} L_1 D_1 D_1^{-1} = D_2 U_2 U_1^{-1} D_1^{-1}$$

$$L_2^{-1} L_1 = D_2 U_2 U_1^{-1} D_1^{-1}$$

Como son matrices triangulares, entonces

$$L_2^{-1} L_1 = D_2 U_2 U_1^{-1} D_1^{-1} = \mathrm{I}$$

$$L_2^{-1} L_1 = \mathrm{I} \qquad \therefore \quad L_2 = L_1$$

De la misma forma, haciendo lo mismo para U_1 y U_2

a partir de $L_1 D_1 U_1 = L_2 D_2 U_2$

Igual para D_1 y D_2

Sistemas que requieren intercambio de renglones

Si al estar calculando U encontramos que es necesario hacer un cambio de renglones, entonces hacer el cambio y empezar de nuevo.
Estos cambios los puede hacer una matriz de “permutación”.

Teorema

Sea A matriz cuadrada de orden n invertible
Entonces existe una matriz de permutación P, una matriz triangular inferior L y una matriz triangular superior U tal que PA=LU

2.14.- Algunas cuestiones de cómputo (pivoteo, escalonamiento y matrices mal-condicionadas)

- Computadoras manejan un número finito de cifras significativas
- Computadoras funcionan con números truncados
  (aunque el despliegue de información si es redondeado)
- Esto puede traer problemas serios

Ejemplo: Suponer una computadora con 3 dígitos significativos

Resta: evaluar 2/3 – 665/1000

0.666... – 0.665 = 0.001666...

pero en computadora 3 dígitos

0.666 – 0.665 = 0.001

Problema común en integración/diferenciación numérica (con sistemas rígidos) .

Pregunta: ¿cuántos dígitos significativos maneja una computadora actualmente?

Pivoteo Parcial

Intercambio de renglones para obtener el valor absoluto más grande del pivote que se va a obtener.

Por ejemplo:

$$\begin{pmatrix} 2 & 8 & -1 & 3 & 4 \\ 0 & -2 & 3 & -5 & 6 \\ 0 & 4 & 1 & 2 & 0 \\ 0 & -7 & 3 & 1 & 4 \end{pmatrix} \rightarrow \begin{pmatrix} 2 & 8 & -1 & 3 & 4 \\ 0 & -7 & 3 & 1 & 4 \\ 0 & 4 & 1 & 2 & 0 \\ 0 & -2 & 3 & -5 & 6 \end{pmatrix}$$

Ejemplo:

$$\begin{aligned} 0.01x_1 + 100x_2 &= 100 \\ -100x_1 + 200x_2 &= 100 \end{aligned}$$

1. Encontrar solución
2. Encontrar solución con Gauss y 3 dígitos significativos
3. Idem anterior pero con pivoteo parcial

$$1.- \quad \left[\begin{array}{cc|c} 0.01 & 100 & 100 \\ -100 & 200 & 100 \end{array}\right] \sim \left[\begin{array}{cc|c} 0.01 & 100 & 100 \\ 0 & 1{,}000{,}200 & 1{,}000{,}100 \end{array}\right]$$

substitución hacia atrás

$$x_2 = \frac{1{,}000{,}100}{1{,}000{,}200} \approx 0.9999$$

$$x_1 = (100 - 100(\frac{1{,}000{,}100}{1{,}000{,}200}))\ 100 \approx 0.9998$$

2.- Solución con 3 dígitos significativos

$$\left[\begin{array}{cc|c} 0.01 & 100 & 100 \\ -100 & 200 & 100 \end{array}\right] \sim \left[\begin{array}{cc|c} 0.01 & 100 & 100 \\ 0 & 1{,}000{,}000 & 1{,}000{,}000 \end{array}\right]$$

sustitución hacia atrás

$x_2 = 1$

$x_1 = (100 – 100)100=0$

Se pierde la información de la 2da columna y vector b. Resultado totalmente diferente

3.- Solución con 3 dígitos significativos y pivoteo parcial

$$\left[\begin{array}{cc|c} 0.01 & 100 & 100 \\ -100 & 200 & 100 \end{array}\right] \sim \left[\begin{array}{cc|c} -100 & 200 & 100 \\ 0.01 & 100 & 100 \end{array}\right]$$

$$\sim \left[\begin{array}{cc|c} -100 & 200 & 100 \\ 0 & 100 & 100 \end{array}\right]$$

sustitución hacia atrás

$x_2 = 100 / 100 = 1$

$x_1 = (100-200) / -100 = 1$

¡Solución más cercana a la primera!

¡Y la información de la segunda columna se mantiene!

¿Por qué funciona?

En caso 2

$$0.01x_1 + 100x_2 = 100$$
$$-100x_1 + 200x_2 = 100$$

En caso 3

$$-100x_1 + 200x_2 = 100$$
$$0.01x_1 + 100x_2 = 100$$

Multiplicar por 10^4
hace que el 200 ya no tenga cabida en las 3 cifras significativas.

Al multiplicar por un número menor que 1 no amenazamos los dígitos significativos de la 2da ecuación

Pivoteo completo

Multiplicar 1era ecuación por 10^4

$$\begin{aligned} 0.01x_1 + 100x_2 &= 100 \\ \underline{-100x_1 + 200x_2} &\underline{= 100} \end{aligned} \longrightarrow \begin{aligned} 100x_1 + 1,000,000x_2 &= 1,000,000 \\ \underline{-100x_1 + \quad 200x_2} &\underline{= \quad 100} \end{aligned}$$

Con pivoteo parcial, no intercambiamos renglones porque hay misma magnitud: 100

Pero resolviendo de esta forma, de nuevo destruimos el 200 de x_2 y estamos en caso 2

$$
\begin{array}{r}
100x_1 + 1,000,000x_2 = 1,000,000 \\
1,000,000x_2 = \qquad 100 \\
\hline
\end{array}
$$

$$x_2 = \frac{1,000,000}{1,000,000} = 1$$

$$x_1 = 0$$

Este problema se evita con

<u>Pivoteo Completo</u>

Si es necesario, columnas intercambian su posición para hacer el pivote en cuestión de la mayor magnitud posible

Entonces para sistema anterior

$$\begin{array}{cc} x_2 & x_1 \\ \end{array}$$

$$\begin{array}{c} \begin{array}{ccc} x_2 & x_1 & \\ \end{array} \\ \left[\begin{array}{cc|c} 1{,}000{,}000 & 100 & 1{,}000{,}000 \\ 200 & -100 & 100 \end{array}\right] \end{array}$$

Notar cambio de orden de variables

El mantener el registro en una computadora consume tiempo.

Resolviendo

$$\left[\begin{array}{cc|c} 1{,}000{,}000 & 100 & 1{,}000{,}000 \\ 0 & -100 & -100 \end{array}\right]$$

$$x_2 = 1$$

$$x_1 = 1$$

Escalamiento

De nuevo con

$$\begin{aligned} 100x_1 + 1{,}000{,}000x_2 &= 1{,}000{,}000 \\ -100x_1 + \quad 200x_2 &= \quad 100 \end{aligned}$$

Observamos que $1{,}000{,}000x_2$ domina peligrosamente la información en la columna x_2

Multiplicar una ecuación por una constante ≡ escalamiento

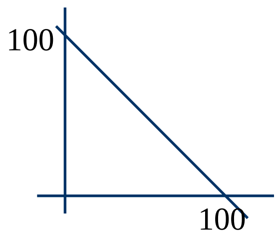
Cuando se hace escalamiento, debido a problemas de redondeo es posible encontrar valores muy pequeños que podrían ser cero.

Algunas implementaciones, cuando calculan pivotes, si se encuentran valores muy pequeños (de magnitud e), se hacen cero.

Es común escalar a magnitudes alrededor de 1 (números adimensionales) y $E=10^{-4}$

Matrices mal condicionadas

Linea $x+y=100$

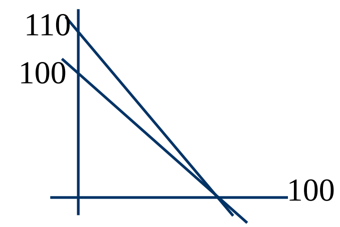


Sistema

$$\left.\begin{aligned} x+y&=100 \\ x+0.9y&=100 \end{aligned}\right\}$$

Resolviendo:

$x=100 \quad y=0$



Ahora, cambiando un poco los coeficientes

$$\left.\begin{aligned} x+y&=100 \\ 0.9x+y&=100 \end{aligned}\right\} \quad \begin{aligned} x&=0 \\ y&=100 \end{aligned}$$

Esto es un ejemplo de un sistema mal-condicionado (o inestable)

- Pequeños cambios en coeficientes de A producen cambios muy grandes en la solución.
- Para un sistema 2x2 podemos ver que el problema es encontrar la solución para 2 líneas casi paralelas.
- Computadoras tienen problemas para resolver estos sistemas debido a los errores de redondeo (pequeños cambios en los coeficientes, cambios grandes en la solución)
- Pivoteo y escalamiento normalmente no ayudan.