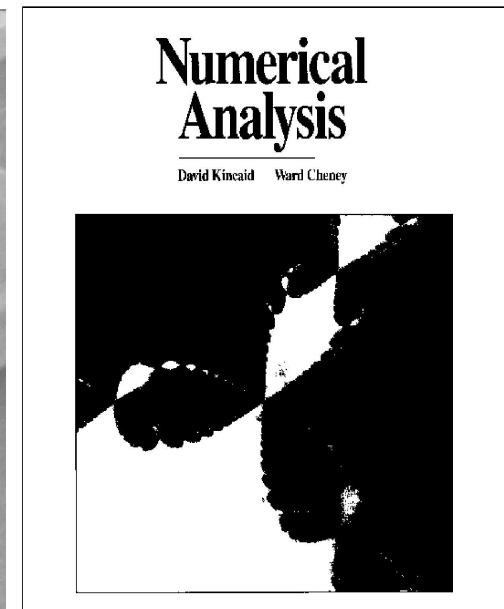
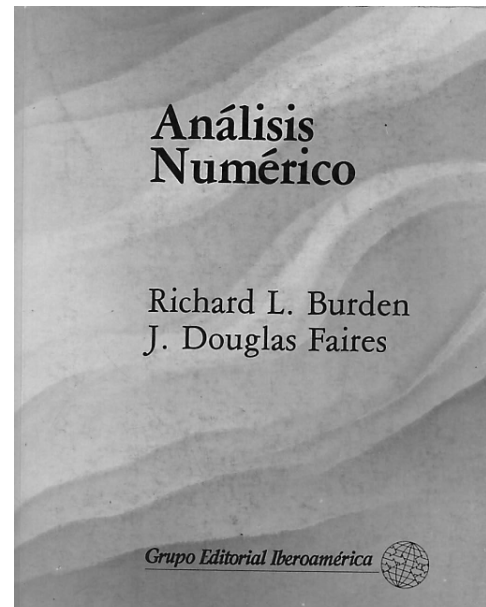


CO3211 – Cálculo Numérico

- Representación de números. Análisis de error.
- Sistemas lineales. Métodos directos e iterativos.
- Autovalores y autovectores.
- Aproximación de Funciones. Mínimos cuadrados. Interpolación. Diferencias divididas. Splines. Curvas paramétricas.

Referencias:

- Análisis Numérico. Richard Burden, Douglas Faires. Grupo Editorial Iberoamericano.
- Numerical Analysis. David Kincaid, Ward Cheney. Brooks-Cole.



CO3211 – Cálculo Numérico

Aula:

Teoría: martes 11:30 a 1:30 - AUL019

jueves 11:30 a 1:30 - AUL103

Laboratorio: viernes 7:30 a 9:30 - MYS Sala A

Evaluación:

2 Parciales con valor de 32 puntos cada uno (semanas 6 y 11)

8 Laboratorios evaluados con valor de 3 puntos cada uno.

1 proyecto con valor de 12 puntos

No hay recuperación de laboratorios evaluados.

Sólo recuperación de 1 parcial (justificado!!) en la semana 12 (toda la materia)

Representación de números

- Representación en base 10

$$0.101_{10} = 1 \cdot 10^{-1} + 0 \cdot 10^{-2} + 1 \cdot 10^{-3} = \frac{1}{10} + \frac{1}{1000} = 0.101$$

- Representación en base 2

$$0.101_2 = 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} = \frac{1}{2} + \frac{1}{8} = 0.625_{10}$$

$$\begin{aligned} 11011.01_2 &= 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2} \\ &= 16 + 8 + 2 + 1 + \frac{1}{4} = 27.25_{10} \end{aligned}$$

En general

$$x = (a_n a_{n-1} \cdots a_1 a_0 . a_{-1} a_{-2} \cdots a_{-m})_2 =$$

$$a_n \cdot 2^n + a_{n-1} \cdot 2^{n-1} + \cdots + a_1 \cdot 2^1 + a_0 \cdot 2^0 + a_{-1} \cdot 2^{-1} + a_{-2} \cdot 2^{-2} + \cdots + a_{-m} \cdot 2^{-m}$$

Representación de números

- Representación en base 2 (cont.)

$$x = (a_n a_{n-1} \cdots a_1 a_0)_2 = a_n \cdot 2^n + a_{n-1} \cdot 2^{n-1} + \cdots + a_1 \cdot 2^1 + a_0 \cdot 2^0 = 2(a_n \cdot 2^{n-1} + a_{n-1} \cdot 2^{n-2} + \cdots + a_1) + a_0$$

Luego a_0 es el resto de dividir x entre 2.

$$x = 2 \cdot x_1 + r_0$$

con

$$x_1 = a_n \cdot 2^{n-1} + a_{n-1} \cdot 2^{n-2} + \cdots + a_1 \quad \text{y} \quad r_0 = a_0$$

Para hallar el siguiente dígito a_1 , aplicamos el mismo procedimiento a x_1

$$x_1 = a_n \cdot 2^{n-1} + a_{n-1} \cdot 2^{n-2} + \cdots + a_1 = 2 \cdot x_2 + r_1$$

con

$$x_2 = a_n \cdot 2^{n-2} + a_{n-1} \cdot 2^{n-3} + \cdots + a_2 \quad \text{y} \quad r_1 = a_1$$

Representación de números

- Representación en base 2 (cont.)

Ejemplo1:

$$x = 25 \Rightarrow x = 2 \cdot 12 + 1 \Rightarrow a_0 = 1$$

$$x_1 = 12 \Rightarrow x_1 = 2 \cdot 6 + 0 \Rightarrow a_1 = 0$$

$$x_2 = 6 \Rightarrow x_2 = 2 \cdot 3 + 0 \Rightarrow a_2 = 0$$

$$x_3 = 3 \Rightarrow x_3 = 2 \cdot 1 + 1 \Rightarrow a_3 = 1$$

$$x_4 = 1 \Rightarrow x_4 = 2 \cdot 0 + 1 \Rightarrow a_4 = 1$$

$$x_5 = 0 \quad \text{finaliza}$$

$$\text{por lo tanto} \quad (25)_{10} = (11001)_2$$

Representación de números

- Representación en base 2 (cont.)

$$x = (a_{-1}a_{-2}\cdots a_{-m})_2 = a_{-1} \cdot 2^{-1} + a_{-2} \cdot 2^{-2} + \cdots + a_{-m} \cdot 2^{-m}$$

$$\frac{1}{2}(a_{-1} + a_{-2} \cdot 2^{-1} + \cdots + a_{-m} \cdot 2^{-m+1})$$

luego $2x = a_{-1} + a_{-2} \cdot 2^{-1} + \cdots + a_{-m} \cdot 2^{-m+1}$

así, a_{-1} es la parte entera de $2x$, y repetimos el mismo proceso para calcular el siguiente dígito

Ejemplo2:

$$x = 0.8125 \Rightarrow 2x = 1.625 \Rightarrow a_{-1} = 1$$

$$x_1 = 0.625 \Rightarrow 2x_1 = 1.25 \Rightarrow a_{-2} = 1$$

$$x_2 = 0.25 \Rightarrow 2x_2 = 0.50 \Rightarrow a_{-3} = 0$$

$$x_3 = 0.50 \Rightarrow 2x_2 = 1.0 \Rightarrow a_{-4} = 1$$

$$x_4 = 0.0 \quad \text{finaliza}$$

por lo tanto $(0.8125)_{10} = (0.1101)_2$

Representación de números

- Representación en base 8

decimal	0	1	2	3	4	5	6	7
binario	000	001	010	011	100	101	110	111
octal	0	1	2	3	4	5	6	7

así, el número se divide en bloques de 3 dígitos (la parte entera de derecha a izquierda y la parte decimal de izquierda a derecha) desde el punto decimal

$$(101|101|001.110|010|100)_2 = (551.624)_8$$

Representación de números

- Representación en base 8 (cont.)

Justificación del cálculo

$$(101101001.1100101)_2 = (101|101|001.110|010|100)_2 = (551.624)_8$$

$$\begin{aligned} x &= (b_{-1}b_{-2}b_{-3}b_{-4}b_{-5}b_{-6}\dots)_2 \\ &= b_{-1} \cdot 2^{-1} + b_{-2} \cdot 2^{-2} + b_{-3} \cdot 2^{-3} + b_{-4} \cdot 2^{-4} + b_{-5} \cdot 2^{-5} + b_{-6} \cdot 2^{-6} + \dots \\ &= (4b_{-1} + 2b_{-2} + b_{-3}) \cdot 2^{-3} + (4b_{-4} + 2b_{-5} + b_{-6}) \cdot 2^{-6} + \dots \\ &= (4b_{-1} + 2b_{-2} + b_{-3}) \cdot 8^{-1} + (4b_{-4} + 2b_{-5} + b_{-6}) \cdot 8^{-2} + \dots \end{aligned}$$

notar que los b_{-i} , $i=1,2,3,\dots$, son los números 0 o 1, y la operación entre paréntesis produce dígitos entre 0 y 7

Representación de números en el computador

Sea β la base usada en el computador, $\beta=2,8,16$.

Sea x un número real, con x distinto de cero y normalizado

$$x = \sigma(a_0.a_1a_2 \cdots a_t)_\beta \beta^E$$

donde

σ representa el signo de x

$(a_1a_2 \cdots a_t)_\beta$ la mantisa en base β

$(E)_\beta$ el exponente en base β

a_0 es un número entero entre 1 y $\beta-1$

Ejemplo de normalización de un número en base 2:

Sea el número $x = 49.8125$

$$x = (110001.1101)_2 = (1.\underbrace{100011101}_2) 2^5$$

5 posiciones

Representación de números en el computador

Una palabra en el computador esta formada por 32 bits.

Así, el número x puede ser representado usando:

- 1 bit para el signo
- 8 bits para el exponente (la característica)
- 23 bits para la parte fraccionaria (la mantisa)

Observación:

El exponente de 8 dígitos representa un número del 0 al $2^8-1 = 255$

$$\begin{array}{cccccccc}
 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & = 1 \cdot 2^7 + 1 \cdot 2^6 + \dots + 1 \cdot 2^1 + 1 \cdot 2^0 = \\
 & & & & & & & & 128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 = 255
 \end{array}$$

A fin de que estos números se puedan representar, se resta 127 del exponente, de manera que el intervalo del exponente es en realidad $[-127, 128]$.

El exponente se almacena sumándole el sesgo correspondiente al número de bits de la representación usada, en este caso el sesgo corresponde a 127.

La razón es que el exponente va ser un número entre -127 y 128.

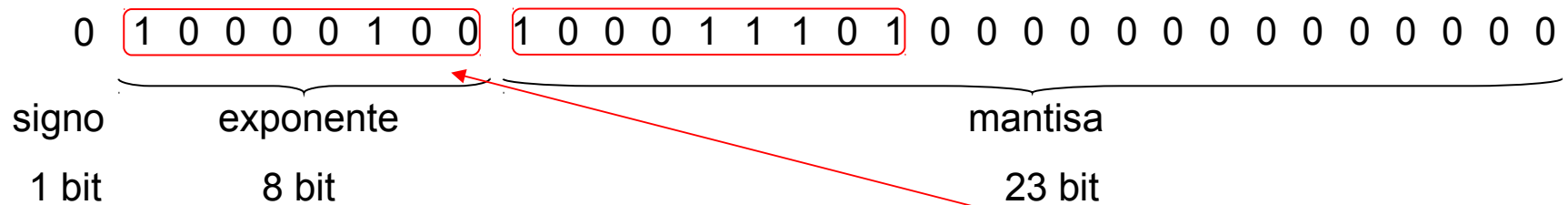
Representación de números en el computador

Retomando el ejemplo $x = 49.8125$

$$x = (110001.1101)_2 = (1.100011101)_2 2^5$$

El número normalizado

$$x = (1.100011101)_2 2^5 \quad \text{se representa como}$$



el exponente $e = (5 + 127 \text{ (sesgo)})_2 : e = 5 + 127 = 132 = (10000100)_2$

el signo s se representa como 0 si es positivo y 1 si es negativo, $s = 0$

la mantisa es $(100011101)_2$, de donde $m = (0.100011101)_2$

El uso de este sistema proporciona un número en punto flotante de la forma

$$(-1)^s 2^{e-127} (1+m)$$

Representación de números en el computador

El número de máquina menor que le sigue a

0 1 0 0 0 0 1 0 0 1 0 0 0 1 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

es

0 1 0 0 0 0 1 0 0 1 0 0 0 1 1 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1

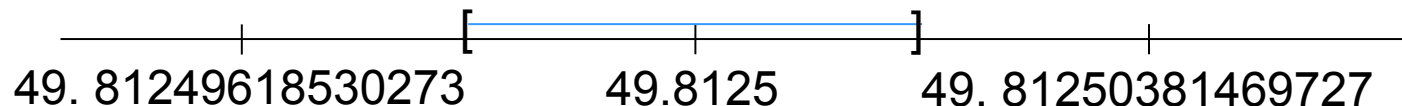
49. 81249618530273

y el siguiente número de máquina mayor es

0 1 0 0 0 0 1 0 0 1 0 0 0 1 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1

49. 81250381469727

Esto significa que nuestro número de máquina original representa no sólo a 49.8125 sino también a los números que están en el intervalo entre corchetes (línea azul).



Para computadores de 64 bits (una palabra), la representación es 1 bit para el signo, 11 bits para el exponente y 52 bits para la mantisa.

Representación de números en el computador

Determinar el número decimal que corresponde al número de máquina (representación de 32 bits) siguiente:

$$(45DE4000)_{16}$$

El número en binario correspondiente es

$$(0100\ 0101\ 1101\ 1110\ 0100\ 0000\ 0000\ 0000)_2$$

0 1 0 0 0 1 0 1 1 1 0 1 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

signo

exponente

mantisa

exponente: $(10001011)_2 = (213)_8 = (139)_{10}$, restando el sesgo: $139 - 127 = 12$

mantiza: 101111001, de donde el número asociado es

$$(1.101111001)_2 \times 2^{12} = (1101111001000.)_2 = (15710)_8 =$$

$$0 \times 1 + 1 \times 8 + 7 \times 8^2 + 5 \times 8^3 + 1 \times 8^4 =$$

$$(7112)_{10}$$

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111

Errores de redondeo y aritmética del computador

El error de redondeo se origina porque la aritmética realizada en un computador involucra números con sólo un número finito de dígitos, con el resultado de que muchos cálculos se realizan con representaciones aproximadas de los números verdaderos.

En un computador, sólo un subconjunto relativamente pequeño de los números reales, los números de punto flotante o números de máquina decimal, se usan para representar a todos los números reales.

Para explicar los problemas que pueden surgir en la manipulación de números de máquina decimal, supondremos que estos se representan como (aritmética de k dígitos)

$$\pm (0.d_1d_2 \cdots d_k) \times 10^n$$

$$\text{con } 1 \leq d_1 \leq 9, \quad 0 \leq d_i \leq 9 \quad \text{para } i = 2, \dots, k$$

Los computadores usan aproximadamente $k = 6$ y $-77 \leq n \leq 76$.

Errores de redondeo y aritmética del computador

Cualquier número real y que este dentro del rango numérico del computador se le puede asociar una representación de punto flotante $fl(y)$.

$$y = (0.d_1d_2 \cdots d_k d_{k+1} \cdots) \times 10^n$$

Hay 2 formas de llevar a cabo esto:

- Truncando: cortar los dígitos d_{k+1}, d_{k+2}, \dots , y así

$$fl(y) = 0.d_1d_2 \cdots d_k \times 10^n$$

- Redondeando: añadir $5 \times 10^{n-(k+1)}$ y luego cortar, obteniéndose

$$fl(y) = 0.\delta_1\delta_2 \cdots \delta_k \times 10^n$$

Este método funciona así:

Si d_{k+1} es mayor o igual a 5, agregamos 1 a d_k

Si d_{k+1} es menor que 5, cortamos los dígitos $k+1$ en adelante

Errores de redondeo y aritmética del computador

Las imprecisiones que resultan de redondear se conocen como

errores por redondeo.

En general, los errores por redondeo se acumulan menos (durante los cálculos repetidos) que los errores producidos por truncamiento, ya que el valor verdadero es mayor que el valor redondeado en alrededor de la mitad de las veces y menor también alrededor de la mitad de las veces.

Además, para la operación de truncamiento, el mayor error absoluto que pudiera producirse sería del doble del de redondeo.

Por otra parte, el truncamiento no requiere de decisión alguna acerca de cambiar o no el último dígito retenido.

Errores de redondeo y aritmética del computador

Ejemplo: usando aritmética de 4 dígitos, realicemos la suma de 2 números para los casos de redondeo y truncamiento:

número	=	redondeo	+	error	=	truncamiento	+	error	
1374.8	=	1375	-	0.2	=	1374	+	0.8	
3856.4	=	3856	+	0.4	=	3856	+	0.4	
Total	5231.2	=	5231	+	0.2	=	5230	+	1.2

Los errores durante el proceso de redondeo tienen signos opuestos y se cancelan parcialmente. Para el caso de truncamiento, sin embargo, los errores tienen el mismo signo y por lo tanto se suman.

Los errores por redondeo se acumulan menos que los errores producidos por truncamiento.

Errores de redondeo y aritmética del computador

Si p^* es una aproximación de p , definimos dos tipos de errores:

- El error absoluto, que viene dado por $EA = |p^* - p|$

- El error relativo, que esta dado por $ER = \frac{|p^* - p|}{|p|}$ siempre y cuando p sea distinto de cero.

Ejemplo:

$$p = 0.3000 \times 10^1$$

$$p^* = 0.3100 \times 10^1$$

$$EA = 0.1$$

$$ER = 0.3333 \times 10^{-1}$$

$$p = 0.3000 \times 10^{-3}$$

$$p^* = 0.3100 \times 10^{-3}$$

$$EA = 0.1 \times 10^{-4}$$

$$ER = 0.3333 \times 10^{-1}$$

Observaciones:

- el error relativo en ambos casos es el mismo, mientras los errores absolutos son diferentes
- como medida de precisión el error absoluto puede ser engañoso, en cambio el error relativo puede ser más significativo

Errores de redondeo y aritmética del computador

Error relativo para la representación de punto flotante $fl(y)$ de y

$$y = (0.d_1d_2 \cdots d_k d_{k+1} \cdots) \times 10^n$$

- Si se usan k dígitos decimales “cortando”

$$fl(y) = 0.d_1d_2 \cdots d_k \times 10^n$$

el error relativo es

$$\begin{aligned} \left| \frac{y - fl(y)}{y} \right| &= \left| \frac{0.d_1d_2 \cdots d_k d_{k+1} \cdots \times 10^n - 0.d_1d_2 \cdots d_k \times 10^n}{0.d_1d_2 \cdots d_k d_{k+1} \cdots \times 10^n} \right| \\ &= \left| \frac{0.d_{k+1} \cdots \times 10^{n-k}}{0.d_1d_2 \cdots d_k d_{k+1} \cdots \times 10^n} \right| = \left| \frac{0.d_{k+1} \cdots}{0.d_1d_2 \cdots d_k d_{k+1} \cdots} \right| 10^{-k} \end{aligned}$$

Como $d_1 \neq 0$, el mínimo valor del denominador es 0.1, y el numerador está acotado por 1, obteniéndose

$$\left| \frac{y - fl(y)}{y} \right| \leq \frac{1}{0.1} \times 10^{-k} = 10^{-k+1}$$

Obs. La cota para el error relativo cuando se usa aritmética de k dígitos, es independiente del número que se está representando.

Errores de redondeo y aritmética del computador

- De manera similar, una cota para el error relativo, cuando se usa aritmética de redondeo de k dígitos,

$$\text{si } d_{k+1} < 5, \text{ entonces } fl(y) = 0.d_1d_2 \cdots d_k \times 10^n$$

$$\text{si } d_{k+1} \geq 5, \text{ entonces } fl(y) = 0.d_1d_2 \cdots d_k \times 10^n + 10^{n-k}$$

Para el segundo caso,

$$\begin{aligned} \left| \frac{y - fl(y)}{y} \right| &= \left| \frac{0.d_{k+1} \cdots \times 10^{n-k} - 10^{n-k}}{0.d_1d_2 \cdots d_k d_{k+1} \cdots \times 10^n} \right| = \left| \frac{0.d_{k+1} \cdots - 1}{0.d_1d_2 \cdots d_k d_{k+1} \cdots} \right| 10^{-k} \\ &\leq \left| \frac{0.d_{k+1} \cdots - 1}{0.1} \right| 10^{-k} = |0.d_{k+1} \cdots - 1| 10^{-k+1} \leq 0.5 \cdot 10^{-k+1} \end{aligned}$$

finalmente

$$\left| \frac{y - fl(y)}{y} \right| \leq 0.5 \cdot 10^{-k+1}$$

Obs. La cota para el error relativo cuando se usa aritmética de k dígitos, es independiente del número que se está representando.

Errores de redondeo y aritmética del computador

Las cotas para el error relativo cuando se usa aritmética de k dígitos, son independientes del número que se está representando.

Esto se debe a la manera en que los “números de máquina decimal” o “números de punto flotante” están distribuidos a lo largo de R .

Debido a la forma exponencial de la característica, se usa la misma cantidad de “números de máquina decimal” para representar cada uno de los intervalos

$$[0.1,1], [1,10], [10,100], \dots, [10^{n-1},10^n]$$

$$[0.1,1] = [0.1,1] \times 10^0 \rightarrow \pm (0.d_1 d_2 \dots d_k) \times 10^0$$

$$[1,10] = [0.1,1] \times 10^1 \rightarrow \pm (0.d_1 d_2 \dots d_k) \times 10^1$$

...

$$[10^{n-1},10^n] = [0.1,1] \times 10^n \rightarrow \pm (0.d_1 d_2 \dots d_k) \times 10^n$$

La cantidad de números es constante para todo entero n .

Errores de redondeo y aritmética del computador

Se dice que el número p^* aproxima a p con t **dígitos significativos** (o cifras), si t es el entero más grande no negativo para el cual

$$\left| \frac{p - p^*}{p} \right| < 5 \cdot 10^{-t}$$

Ejemplos:

Supongamos que p^* aproxima a 1000 al menos con 4 cifras significativas, entonces

$$\left| \frac{1000 - p^*}{1000} \right| < 5 \cdot 10^{-4}$$

$$\Leftrightarrow -0.5 < p^* - 1000 < 0.5$$

$$\Leftrightarrow 999.5 < p^* < 1000.5$$

Supongamos que p^* aproxima a 5000 al menos con 4 cifras significativas, entonces

$$\left| \frac{5000 - p^*}{5000} \right| < 5 \cdot 10^{-4}$$

$$\Leftrightarrow -2.5 < p^* - 5000 < 2.5$$

$$\Leftrightarrow 4997.5 < p^* < 5002.5$$

Errores de redondeo y aritmética del computador

La tabla siguiente ilustra la naturaleza continua del concepto de **dígitos significativos**, listando, para varios valores de p , la mínima cota superior de

$$|p - p^*|$$

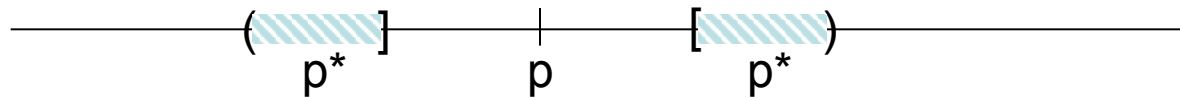
cuando p^* concuerda con p en cuatro cifras significativas

p	0.1	0.5	100	1000	5000	9990	10000
$máx p - p^* $	0.00005	0.00025	0.05	0.5	2.5	4.995	5

Errores de redondeo y aritmética del computador

Si el número p^* tiene exactamente 4 cifras significativas respecto a otro número p , determinar la región donde está ubicado p^* .

$$\left| \frac{p - p^*}{p} \right| < 5 \cdot 10^{-4} \quad \text{y} \quad \left| \frac{p - p^*}{p} \right| \geq 5 \cdot 10^{-5}$$



$$\left| \frac{p - p^*}{p} \right| < 5 \cdot 10^{-4} \quad \Rightarrow \quad p^* \in (p - |p|5 \cdot 10^{-4}, p + |p|5 \cdot 10^{-4})$$

$$\left| \frac{p - p^*}{p} \right| \geq 5 \cdot 10^{-5} \quad \Rightarrow \quad p^* \in (-\infty, p - |p|5 \cdot 10^{-5}] \cup [p + |p|5 \cdot 10^{-5}, +\infty)$$

Finalmente

$$p^* \in (p - |p|5 \cdot 10^{-4}, p - |p|5 \cdot 10^{-5}] \cup [p + |p|5 \cdot 10^{-5}, p + |p|5 \cdot 10^{-4})$$

Representación de números en el computador

Además de tener una representación inexacta de los números reales, la aritmética realizada en el computador no es exacta.

Supongamos que \oplus , \ominus , \otimes y \oslash representan las operaciones de suma, resta, multiplicación y división en el computador (aritmética idealizada):

$$x \oplus y = fl(fl(x) + fl(y))$$

$$x \ominus y = fl(fl(x) - fl(y))$$

$$x \otimes y = fl(fl(x) \times fl(y))$$

$$x \oslash y = fl(fl(x) / fl(y))$$

Representación de números en el computador

Ejemplo:

Consideremos la siguiente suma:

$$0.99 + 0.0044 + 0.0042.$$

Con aritmética exacta, el resultado es 0.9986.

Sin embargo, si usamos una aritmética de tres dígitos, y las operaciones se realizan siguiendo el orden de izquierda a derecha, encontramos que

$$(0.99+0.0044)+0.0042 = 0.994+0.0042 = 0.998.$$

Por otra parte, si operamos primero los dos últimos números, tenemos que:

$$0.99+(0.0044+0.0042) = 0.99+0.0086 = 0.999,$$

lo cual demuestra el efecto del error por redondeo en un caso tan simple como éste.

(¿Por qué sucede esto?).

Desde el punto de vista numérico es importante el orden en que sumamos.

Representación de números en el computador

Ejemplo: usando aritmética de cinco dígitos truncando, para $x = 1/3$, $y = 5/7$

	resultado	valor real	error absoluto	error relativo
$x \oplus y$	0.10476×10^1	$22/21$	0.190×10^{-4}	0.182×10^{-4}
$x \ominus y$	0.38095×10^0	$8/21$	0.238×10^{-5}	0.625×10^{-5}
$x \otimes y$	0.23809×10^0	$5/27$	0.524×10^{-5}	0.220×10^{-4}
$x \oslash y$	0.21428×10^1	$15/7$	0.571×10^{-4}	0.267×10^{-4}

Error relativo máximo obtenido 0.267×10^{-4} , para lo cual la aritmética de cinco dígitos cortando produce resultados satisfactorios.

Ejemplo: usando aritmética de cinco dígitos truncando, para $x = 1/3$, $y = 5/7$, $u = 0.714251$, $v = 98765.9$, $w = 0.11111 \times 10^{-4}$

	resultado	valor real	error absoluto	error relativo
$y \ominus u$	0.30000×10^{-4}	0.34714×10^{-4}	0.471×10^{-5}	0.136×10^0
$(y \ominus u) \oslash w$	0.27000×10^1	0.31243×10^1	0.424×10^0	0.136×10^0
$(y \ominus u) \otimes v$	0.29629×10^1	0.34285×10^1	0.465×10^0	0.136×10^0
$u \oplus v$	0.98765×10^5	0.98766×10^5	0.161×10^1	0.163×10^{-4}

Pueden ocurrir errores significativos

Representación de números en el computador

Operaciones que producen errores:

- la suma de muchos números siempre genera errores de redondeo, lo importante es que estos errores no inutilicen el resultado!
- la división de un número con dígitos finitos entre un número de magnitud muy pequeña
- la multiplicación de un número con dígitos finitos por un número relativamente grande
- la sustracción de números casi iguales

$$fl(x) = 0.a_1a_2 \cdots a_p a_{p+1} \cdots a_k \times 10^n \quad fl(y) = 0.a_1a_2 \cdots a_p b_{p+1} \cdots b_k \times 10^n$$

$$fl(fl(x) - fl(y)) = 0.d_{p+1} \cdots d_k \times 10^{n-p}$$

así, el número usado para representar $x-y$ tendrá sólo $k-p$ cifras significativas (en la mayoría de los computadores, se le asignarán k dígitos, pero los últimos p serán asignados al azar).

Cualquier cálculo adicional que involucre $x-y$ retendría el problema de tener solamente $k-p$ cifras significativas.

Con el objeto de evitar esta dificultad se recomienda buscar una reformulación del algoritmo inicialmente considerado.

Representación de números en el computador

Ejemplo:

Consideremos $x^2 + 62.10x + 1 = 0$, cuyas raíces tienen los valores aproximados

$$x_1 \approx -0.01610723 \quad \text{y} \quad x_2 \approx -62.08390$$

En esta ecuación, b^2 es mucho mayor que 4, de modo que el numerador en el cálculo de x_1 implica la resta de números casi iguales.

Ahora bien, si suponemos una aritmética de redondeo a cuatro cifras, y como

$$\Delta = \sqrt{b^2 - 4} = \sqrt{3856.0 - 4.000} = \sqrt{3852.} = 62.06,$$

tenemos que

$$fl(x_1) = \frac{-62.10 + 62.06}{2.000} = \frac{-0.0400}{2.000} = -0.0200,$$

la cual es una mala aproximación a $x_1 = 0.01611$, con un error relativo grande de $\approx 2.4 \times 10^{-1}$.

Representación de números en el computador

Ejemplo (cont.):

Esta dificultad puede evitarse racionalizando el numerador en la fórmula cuadrática

$$x_1 = \frac{(\Delta - b)(\Delta + b)}{2(\Delta + b)} = \frac{(\Delta^2 - b^2)}{2(\Delta + b)} = \frac{-4}{2(\Delta + b)} = \frac{-2}{(\Delta + b)}.$$

Se obtiene en este caso

$$x_1 = \frac{-2.000}{62.10 + 62.06} = \frac{-2.000}{124.2} = -0.01610,$$

con un error relativo de $\approx 6.2 \times 10^{-4}$, mucho menor que el anterior.

Por otro lado, el cálculo de x_2 implica la suma de dos números casi iguales.

$$-b \quad \text{y} \quad -\sqrt{b^2 - 4}.$$

Pero esto no representa problema alguno, pues

$$fl(x_2) = \frac{-62.10 - 62.06}{2.000} = \frac{-124.2}{2.000} = -62.10,$$

que tiene un error relativo pequeño de $\approx 3.2 \times 10^{-4}$.

Algoritmos y convergencia

Un algoritmo es un procedimiento que describe, sin ninguna ambigüedad, una sucesión finita de pasos a realizar en un orden específico.

El objetivo de un algoritmo será generalmente el de implantar un procedimiento numérico para resolver un problema o aproximar una solución del problema.

Como vehículo para describir algoritmos usaremos un “pseudocódigo”.

Los pasos en los algoritmos se arreglan de tal manera que la dificultad de traducir en un lenguaje de programación (MATLAB, FORTRAN, C, ...) apropiado para aplicaciones científicas sea mínimo.

Algoritmos y convergencia

Ejemplo:

algoritmo para calcular la suma de los números x_1 al x_n , es decir,

$$\sum_{i=1}^n x_i = x_1 + x_2 + \cdots + x_n$$

donde n y los números x_1 al x_n están dados.

Entrada: n, x_1, x_2, \dots, x_n

P1: sum = 0

P2: para $i = 1$ hasta n

sum = sum + x_i

fin para

P3: escribir sum

P4: parar

Obs. Considerar la suma

$$s = 9.87 + 0.78 + 0.05 + 0.01$$

Al realizar esta operación de izquierda a derecha y de derecha a izquierda usando aritmética de 3 dígitos con redondeo, se obtienen los resultados:

10.8 y 10.7

Algoritmos y convergencia

Un **algoritmo** es una secuencia finita de operaciones algebraicas y lógicas que producen una solución aproximada de un problema matemático

Análisis Numérico → diseño de algoritmos y estudio de su eficiencia

Eficiencia →

- requerimiento de memoria,
- tiempo de cálculo (rapidez)
- estimación del error (precisión)

$$\begin{array}{ccccccc}
 \text{errores de} & & & & & & \text{errores de} \\
 \text{entrada} & & & & & & \text{de} \\
 \text{(en medidas)} & + & \text{errores de} & + & \text{errores de} & = & \text{salida} \\
 & & \text{almacenamiento} & & \text{algoritmo} & & \\
 & & \underbrace{\hspace{10em}} & & & & \\
 & & \text{a analizar} & & & &
 \end{array}$$

Algoritmos y convergencia

Los **errores** se propagan a través de los cálculos, debido a la estructura propia del algoritmo. Para estudiar esta propagación y por lo tanto el error final, atendemos a los conceptos de condicionamiento y estabilidad.

Condicionamiento: mide la influencia que tendrían los errores en los datos en el caso en que se pueda trabajar con aritmética exacta. No depende del algoritmo sino del problema en si.

Estabilidad: está relacionada con la influencia que tienen en los resultados finales la acumulación de errores que se producen al realizar las diferentes operaciones elementales que constituyen el algoritmo.

condicionamiento y estabilidad → permite estudiar la precisión de un algoritmo para un problema concreto

Algoritmos y convergencia

Condicionamiento

Diremos que un problema está mal condicionado cuando pequeños cambios en los datos dan lugar a grandes cambios en las respuestas. Para estudiar el condicionamiento de un problema se introduce el llamado número de condición de ese problema, específico del problema, que es mejor cuando más cerca de 1 (el problema está bien condicionado) y peor cuando más grande sea (mal condicionado).

La gravedad de un problema mal condicionado reside en que su resolución puede producir soluciones muy dispares en cuanto los datos cambien muy poco.

Estabilidad

Todo algoritmo que resuelve un problema numéricamente produce en cada paso un error numérico.

Un algoritmo se dice inestable cuando los errores que se cometen en cada etapa del mismo van aumentando de forma progresiva, de manera que el resultado final pierde gran parte de su exactitud.

Un algoritmo es estable cuando no es inestable (está controlado).

Algoritmos y convergencia

Ejemplo:

sistema 1

$$\begin{aligned} 10x_1 + 7x_2 + 8x_3 + 7x_4 &= 32 \\ 7x_1 + 5x_2 + 6x_3 + 5x_4 &= 23 \\ 8x_1 + 6x_2 + 10x_3 + 9x_4 &= 33 \\ 7x_1 + 5x_2 + 9x_3 + 10x_4 &= 31 \end{aligned}$$



Sistema mal condicionado

$$\begin{aligned} x_1 &= 1 \\ x_2 &= 1 \\ x_3 &= 1 \\ x_4 &= 1 \end{aligned}$$

sistema 2

$$\begin{aligned} 10x_1 + 7x_2 + 7.983x_3 + 7.019x_4 &= 32 \\ 7.08x_1 + 5.04x_2 + 6x_3 + 5x_4 &= 23 \\ 8x_1 + 5.98x_2 + 9.89x_3 + 9x_4 &= 33 \\ 6.99x_1 + 4.99x_2 + 9x_3 + 9.98x_4 &= 31 \end{aligned}$$



$$x_1 = -81, x_2 = 137, x_3 = -34, x_4 = 22$$

sistema 3

$$\begin{aligned} 10x_1 + 7x_2 + 8x_3 + 7x_4 &= 32.1 \\ 7x_1 + 5x_2 + 6x_3 + 5x_4 &= 22.9 \\ 8x_1 + 6x_2 + 10x_3 + 9x_4 &= 32.98 \\ 7x_1 + 5x_2 + 9x_3 + 10x_4 &= 31.02 \end{aligned}$$



$$x_1 = 7.28, x_2 = -9.36, x_3 = 3.54, x_4 = -0.5$$

Pequeños cambios en los datos en algunos elementos producen grandes cambios en las soluciones:

- entre los sistemas 1 y 2 cambios del orden de 11 centésimas (en la matriz) producen variaciones de hasta 136 unidades en la solución,
- entre los sistemas 1 y 3 cambios del orden de 1 décima (en el término de la derecha) producen variaciones de hasta de 10 unidades en la solución.

Algoritmos y convergencia

Estabilidad

Supongamos que E_n representa el crecimiento del error después de n operaciones subsecuentes.

Si $|E_n| \approx c n + \varepsilon$, donde c es una constante independiente de n , diremos que el crecimiento del error es lineal.

Si $|E_n| \approx k^n \varepsilon$, para $k > 1$, el crecimiento del error es exponencial.

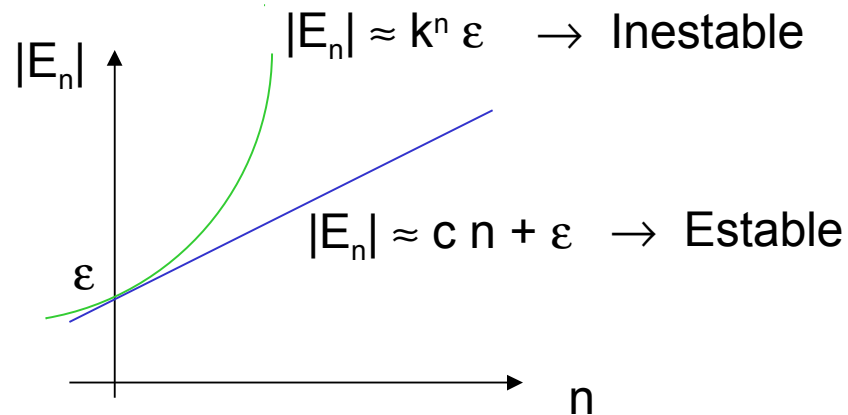
- El crecimiento del error es inevitable, y cuando el crecimiento es lineal y c y ε son pequeños, los resultados son generalmente aceptables.
- El crecimiento exponencial del error debe ser evitado, ya que el término k^n será grande aún para valores relativamente pequeños de ε .

Algoritmos y convergencia

Estabilidad

Consecuencias:

- Un algoritmo que exhibe crecimiento lineal del error es ESTABLE
- Un algoritmo en el que el crecimiento del error es exponencial es INESTABLE



Algoritmos y convergencia

Diremos que un software o subprograma es **robusto** si es capaz de enfrentarse a una amplia variedad de situaciones numéricas distintas sin la intervención del usuario.

Es un hecho conocido que cuando estudiamos diferentes métodos del cálculo científico se presenta una interdependencia entre la **velocidad** y la **confiabilidad** (la primera está directamente relacionada con los costos).

Así, para algunos problemas que involucran cálculo intensivo (como la solución numérica de ecuaciones en derivadas parciales), la velocidad es lo más importante.

Sin embargo, cuando nos referimos a un software de propósito general para ser utilizado por una amplia gama de usuarios, la confiabilidad y la robustez son las características que marcan la pauta.

Por muchos años se han realizado esfuerzos por generar paquetes numéricos de propósito general que reúnan ambas características, confiabilidad y robustez.

Algoritmos y convergencia

El siguiente ejemplo muestra lo que puede suceder cuando escogemos un algoritmo inadecuado. El error por redondeo puede acabar completamente con el resultado de un cálculo.

Supongamos que queremos estimar para $n = 0, 1, \dots, 8$, la integral

$$y_n = \int_0^1 \frac{x^n}{x+5} dx.$$

Observamos que

$$y_n + 5y_{n-1} = \int_0^1 \frac{x^n + 5x^{n-1}}{x+5} dx = \int_0^1 \frac{x^{n-1}(x+5)}{x+5} dx = \frac{1}{n}.$$

Supongamos que en nuestros cálculos usamos sólo tres dígitos

$$y_0 = \int_0^1 \frac{1}{x+5} dx = \ln(x+5) \Big|_0^1 \approx 0.182.$$

$$\text{error en este cálculo } \delta \doteq \left| \frac{p-p^*}{p} \right| < 0.5 \cdot 10^{-3} = 5 \cdot 10^{-4}$$

Algoritmos y convergencia

Observamos que la sucesión es decreciente y todos sus términos son positivos

$$y_n \geq y_{n+1} \quad \text{y} \quad y_n \geq 0$$

$$x \in [0,1]$$

$$n \in \mathbb{N} \Rightarrow x^n \geq x^{n+1} \Rightarrow \frac{x^n}{x+5} \geq \frac{x^{n+1}}{x+5}$$

$$\Rightarrow \int_0^1 \frac{x^n}{x+5} dx \geq \int_0^1 \frac{x^{n+1}}{x+5} dx \Rightarrow y_n \geq y_{n+1}$$

$$x \in [0,1]$$

$$\frac{x^n}{x+5} \geq 0$$

así, la integral de una función positiva es positiva, es decir,

$$y_n = \int_0^1 \frac{x^n}{x+5} dx \geq 0$$

Algoritmos y convergencia

- Consideremos el algoritmo

$$y_1 = 1 - 5y_0 = 1 - 0.910 \approx 0.090$$

$$y_2 = \frac{1}{2} - 5y_1 \approx 0.050$$

$$y_3 = \frac{1}{3} - 5y_2 \approx 0.083 \quad (\text{¡}y_3 > y_2\text{!})$$

$$y_4 = \frac{1}{4} - 5y_3 \approx -0.165 \quad \text{¡sin sentido!}$$

La causa de este resultado está en que el error por redondeo δ en y_0 (cuya magnitud es del orden de 5×10^{-4}) se multiplica por -5 en el cálculo de y_1 , el cual tendrá entonces un error de -5δ .

Ese error produce, a su vez, un error en y_2 de 25δ , en y_3 de -125δ , y en y_4 de 625δ (en el que el error será tan grande como $625 \times 5 \times 10^{-4} = 0.3125$).

Si usáramos una mayor precisión, con más lugares decimales, los resultados “sin sentido” aparecerán en una etapa posterior.

Este fenómeno (por supuesto, indeseable) se conoce como **inestabilidad numérica**. La inestabilidad numérica puede corregirse si encontramos un algoritmo más adecuado.

Algoritmos y convergencia

- Consideremos el algoritmo

$$y_{n-1} = \frac{1}{5n} - \frac{y_n}{5}.$$

En este caso el error estaría, en cada paso, dividido por -5 (aunque necesitaremos un valor de entrada).

$$y_9 + 5y_9 = \frac{1}{10} \Rightarrow y_9 \approx \frac{1}{60} \approx 0.017$$

Observemos directamente de la definición de y_n , que la misma decrece cuando n aumenta.

$$y_8 = \frac{1}{45} - \frac{y_9}{5} \approx 0.019$$

Por lo que podemos asumir que, cuando n es grande, $y_{n+1} \approx y_n$.

$$y_7 = \frac{1}{40} - \frac{y_8}{5} \approx 0.021$$

De esta manera, podemos por conveniencia suponer que $y_{10} \approx y_9$, de donde sigue que

$$y_6 \approx 0.025, \quad y_5 \approx 0.028,$$

$$y_4 \approx 0.034, \quad y_3 \approx 0.043,$$

$$y_2 \approx 0.058, \quad y_1 \approx 0.088,$$

$$y_0 \approx 0.182$$

¡bueno!

Sistemas de Ecuaciones Lineales

Matrices, vectores y escalares:

- Una **matriz** A de dimensión $m \times n$ es un arreglo rectangular de números de la forma

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1,n-1} & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2,n-1} & a_{2n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{m-1,1} & a_{m-1,2} & \cdots & a_{m-1,n-1} & a_{m-1,n} \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n-1} & a_{m,n} \end{pmatrix} = (a_{ij})$$

Se escribe $A \in R^{m \times n}$. Si $m = n$, tal que A es cuadrada, se dice que A es de orden n .

- Los números a_{ij} se denominan los elementos de la matriz A . Por convención el índice i , denominado índice de filas, indica la fila en la cual el elemento está. El otro índice, j , llamado índice de columnas, indica la columna en la cual el elemento está.

Sistemas de Ecuaciones Lineales

Matrices, vectores y escalares (cont.):

- Un **vector** x de dimensión n es un arreglo de la forma

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = (x_j)$$

Se escribe $x \in R^n$. Los números x_j se denominan las componentes de x .

- Por convención todos los vectores son vectores columnas, sus componentes forman una columna. Objetos como $(x_1 \ x_2 \ \cdots \ x_n)$ cuyos componentes forman una fila, se denominan vectores filas. Escribiremos los vectores filas como x^T , la transpuesta de x .
- No haremos distinción alguna entre $R^{n \times 1}$ y R^n , una matriz de dimensión $n \times 1$ y un vector de dimensión n . De igual manera será lo mismo el conjunto de todos los números reales R , también denominados **escalares** y el conjunto de los vectores de dimensión 1 y las matrices de dimensión 1×1 .

Sistemas de Ecuaciones Lineales

Operaciones con matrices:

- Multiplicación de una matriz A por un escalar μ

$$\mu A = \mu (a_{ij}) = (\mu a_{ij})$$

- Suma de matrices A y B de igual dimensión

$$A + B = (a_{ij}) + (b_{ij}) = (a_{ij} + b_{ij})$$

- Matriz nula es la que cuyos elementos son todos ceros, se denota por 0

$$A + 0 = 0 + A = A$$

- Producto de matrices: sea A una matriz $l \times m$ y B una matriz $m \times n$, el producto de A y B es

$$AB = (a_{ik})(b_{kj}) = \left(\sum_{k=1}^m a_{ik} b_{kj} \right)$$

Notar que para que el producto de A y B este definido, el número de columnas de A debe ser igual al número de filas de B .

Un caso particular es el producto matriz-vector

$$AX = (a_{ik})(b_k) = \left(\sum_{k=1}^n a_{ik} b_k \right)$$

Sistemas de Ecuaciones Lineales

Operaciones con matrices (cont.):

- **Matriz identidad** I_n de orden n

$$I_n = \begin{pmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & \cdots & 0 & 1 \end{pmatrix}$$

Si A es una matriz $m \times n$, es fácil verificar que $I_m A = A I_n$. Cuando en el contexto es claro el orden de la matriz identidad, se omite el índice de esta.

- **Matriz diagonal:** Una matriz D es diagonal si todos sus elementos que están fuera de la diagonal son nulos, es decir

$$d_{ij} = 0 \quad \text{siempre y cuando } i \neq j$$

Escribiremos $D = \text{diag}(d_1, d_2, \dots, d_n)$, donde los d_1, d_2, \dots, d_n son los elementos de la diagonal de D .

Sistemas de Ecuaciones Lineales

Operaciones con matrices (cont.):

- En muchas ocasiones es útil escribir el sistema de ecuaciones

$$b_1 = a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n$$

$$b_2 = a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n$$

⋮

$$b_n = a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n$$

en forma abreviada como $b = Ax$, donde A es una matriz cuadrada de orden n , y b y x son vectores de dimensión n .

- Además, se tiene que la suma y producto de matrices son asociativos

$$(A + B) + C = A + (B + C) \quad \text{y} \quad (A B)C = A (B C),$$

el producto es distributivo respecto a la suma

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

la suma de matrices es conmutativa

$$A + B = B + A,$$

Sistemas de Ecuaciones Lineales

Operaciones con matrices (cont.):

El producto de matrices no es conmutativo, en general se tiene

$$A B \neq B A,$$

cuando estos producto están bien definidos.

- **Transpuesta de una matriz:** si A es una matriz de dimensión $m \times n$, se define la matriz transpuesta de A como

$$\text{si } A = (a_{ij}), \text{ entonces } A^t = (a_{ji})$$

La transpuesta es la matriz que se obtiene reflejando la matriz a través de la diagonal principal.

Obs. Sean las matrices A $m \times p$ y B $p \times n$, entonces $(AB)^t = B^t A^t$.

- Si x e y son vectores de dimensión n , entonces

$$y^t x = x_1 y_1 + x_2 y_2 + \cdots + x_n y_n$$

es un escalar denominado el **producto interno** de x e y .

Como resultado inmediato se tiene que $x^t x = x_1 x_1 + \cdots + x_n x_n \geq 0$

Sistemas de Ecuaciones Lineales

Operaciones con matrices (cont.):

- **Desigualdad de Cauchy-Schwarz**

Si x e y son vectores de dimensión n , entonces se verifica que

$$|y^t x|^2 \leq (x^t x) (y^t y)$$

Prueba.

Sean x e y son vectores de dimensión n y λ un escalar cualquiera.

La desigualdad es trivial en el caso $y = 0$. Así suponemos que $y \neq 0$, entonces

$$0 \leq (x - \lambda y)^t (x - \lambda y) = x^t x - \lambda y^t x - \lambda x^t y + \lambda^2 y^t y$$

$$0 \leq x^t x - 2\lambda y^t x + \lambda^2 y^t y$$

Tomando $\lambda = (y^t x) (y^t y)^{-1}$ se obtiene

$$0 \leq x^t x - 2\lambda y^t x + \lambda^2 y^t y = x^t x - (y^t x)^2 (y^t y)^{-1}$$

De donde

$$|y^t x|^2 \leq (x^t x) (y^t y)$$



Sistemas de Ecuaciones Lineales

Operaciones con matrices (cont.):

- **Matrices triangular superior e inferior:**

si A es una matriz cuadrada de dimensión n ,

$A = (a_{ij})$ se denomina triangular superior si $a_{ij} = 0$ para los $i > j$.

$A = (a_{ij})$ se denomina triangular inferior si $a_{ij} = 0$ para los $i < j$.

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1,n-1} & a_{1n} \\ 0 & a_{22} & \cdots & a_{2,n-1} & a_{2n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & a_{n-1,n-1} & a_{n-1,n} \\ 0 & 0 & \cdots & 0 & a_{n,n} \end{pmatrix} \quad \text{triangular superior}$$

$$A = \begin{pmatrix} a_{11} & 0 & \cdots & 0 & 0 \\ a_{21} & a_{22} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n-1,1} & a_{n-1,2} & \cdots & a_{n-1,n-1} & 0 \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n-1} & a_{n,n} \end{pmatrix} \quad \text{triangular inferior}$$

Sean A y B matrices cuadradas de dimensión n .

Si A y B son matrices triangular superior, ¿qué se puede decir del producto de A y B ?

Sistemas de Ecuaciones Lineales

Operaciones con matrices (cont.):

- **Matrices por bloque:**

Usualmente es útil crear una partición de una matriz como una colección de submatrices. Por ejemplo

$$A = \begin{pmatrix} 1 & 2 & -1 \\ 3 & -4 & -3 \\ 6 & 5 & 0 \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \quad y \quad B = \begin{pmatrix} 2 & -1 & 7 & 0 \\ 3 & 0 & 4 & 5 \\ -2 & 1 & -3 & 1 \end{pmatrix} = \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix}$$

Para el producto de las matrices podemos proceder como

$$AB = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix} = \begin{pmatrix} A_{11}B_{11} + A_{12}B_{21} & A_{11}B_{12} + A_{12}B_{22} \\ A_{21}B_{11} + A_{22}B_{21} & A_{21}B_{12} + A_{22}B_{22} \end{pmatrix}$$

siempre y cuando los productos con las submatrices tengan sentido.

Si ahora se crea una partición de B como

$$B = \begin{pmatrix} 2 & -1 & 7 & 0 \\ 3 & 0 & 4 & 5 \\ -2 & 1 & -3 & 1 \end{pmatrix} = \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix}$$

¿el resultado anterior para AB sigue siendo válido?

Sistemas de Ecuaciones Lineales

Operaciones con matrices (cont.):

- **Matrices por bloque.** ejemplo:

$$\left[\begin{array}{c} [1 \quad 2] \\ [-1 \quad 1] \\ [0 \quad 1] \\ [1 \quad -1] \\ [1 \quad 0] \end{array} \right] \left[\begin{array}{c} [1 \quad -1 \quad 0 \quad 1] \\ [1 \quad 0 \quad -1 \quad 1] \\ [-1 \quad 1 \quad 0 \quad 1] \\ [0 \quad 0 \quad 1 \quad 0] \\ [1 \quad 2 \quad 1 \quad 0] \end{array} \right] \left[\begin{array}{c} [1 \quad 0 \quad 1] \\ [-1 \quad 1 \quad 2] \\ [1 \quad 0 \quad 1] \\ [-1 \quad 1 \quad 0] \\ [2 \quad 1 \quad 0] \\ [0 \quad 1 \quad 1] \end{array} \right] \left[\begin{array}{c} [2 \quad 1] \\ [0 \quad 1] \\ [1 \quad 2] \\ [0 \quad 1] \\ [-2 \quad 1] \\ [-1 \quad 1] \end{array} \right]$$

$$= \left[\begin{array}{c} [1 \quad 2 \quad 7] \\ [-3 \quad 1 \quad 3] \\ [-3 \quad 3 \quad 2] \\ [4 \quad 0 \quad -1] \\ [2 \quad 3 \quad 2] \end{array} \right] \left[\begin{array}{c} [2 \quad 5] \\ [0 \quad 2] \\ [-2 \quad 1] \\ [0 \quad 1] \\ [1 \quad 6] \end{array} \right]$$

$$C = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix} = \begin{pmatrix} A_{11}B_{11} + A_{12}B_{21} & A_{11}B_{12} + A_{12}B_{22} \\ A_{21}B_{11} + A_{22}B_{21} & A_{21}B_{12} + A_{22}B_{22} \end{pmatrix}$$

$$A_{11}B_{11} + A_{12}B_{21} = C_{11} \Rightarrow [1 \quad 2] \begin{bmatrix} 1 & 0 & 1 \\ -1 & 1 & 2 \end{bmatrix} + [1 \quad -1 \quad 0 \quad 1] \begin{bmatrix} 1 & 0 & 1 \\ -1 & 1 & 0 \\ 2 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} = [1 \quad 2 \quad 7]$$

Sistemas de Ecuaciones Lineales

Operaciones con matrices (cont.):

- **Matrices por bloque:**

Para la traspuesta de una matriz por bloques podemos proceder como

$$A = \begin{pmatrix} 1 & 2 & -1 \\ 3 & -4 & -3 \\ 6 & 5 & 0 \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \Rightarrow A^t = \begin{pmatrix} 1 & 3 & 6 \\ 2 & -4 & 5 \\ -1 & -3 & 0 \end{pmatrix} = \begin{pmatrix} A_{11}^t & A_{21}^t \\ A_{12}^t & A_{22}^t \end{pmatrix}$$

$$A = \begin{pmatrix} 1 & 2 & 0 & 0 \\ 3 & 4 & 0 & 0 \\ 0 & 0 & 2 & 3 \\ 0 & 0 & 4 & 5 \end{pmatrix} = \begin{pmatrix} U & 0 \\ 0 & V \end{pmatrix} \Rightarrow A^t = \begin{pmatrix} 1 & 3 & 0 & 0 \\ 2 & 4 & 0 & 0 \\ 0 & 0 & 2 & 4 \\ 0 & 0 & 3 & 5 \end{pmatrix} = \begin{pmatrix} U^t & 0 \\ 0 & V^t \end{pmatrix}$$

matrices nula

Sistemas de Ecuaciones Lineales

Operaciones con matrices (cont.):

- **Radio espectral de una matriz:**

si A es una matriz real de dimensión $n \times n$ y $\lambda_1, \dots, \lambda_n$ los autovalores de A , se define el radio espectral de A como

$$\rho(A) = \max_{1 \leq i \leq n} \{|\lambda_i|\}.$$

El espectro de A es el conjunto $\{\lambda_1, \dots, \lambda_n\}$ de los autovalores de A .

Sea λ un escalar (real o complejo), si la ecuación $Ax = \lambda x$, tiene una solución no trivial (esto es, $x \neq 0$), entonces λ es un autovalor de A .

Un vector no cero x que satisfaga la ecuación anterior, es el autovector de A correspondiente al autovalor λ .

Ejemplo:

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

-2 es un autovalor de la matriz 3×3 dada, y el vector $(1, 3, -4)^T$ es el autovector correspondiente.

Sistemas de Ecuaciones Lineales

Operaciones con matrices (cont.):

- **Rango de una matriz.**

si A es una matriz real de dimensión $n \times n$, el rango de A es la dimensión del espacio generado por los vectores columnas de A .

Este se denota como $\text{rank}(A)$.

La matriz A se denomina de rango completo cuando $\text{rank}(A) = n$

Sistemas de Ecuaciones Lineales

Norma de vectores y matrices:

- Sobre los elementos de R^n , el espacio de los vectores de dimensión n , definimos una norma como una función $\| \cdot \|$ de R^n en R^+ que cumple

$$\|x\| \geq 0 \quad \text{para todo } x \in R^n$$

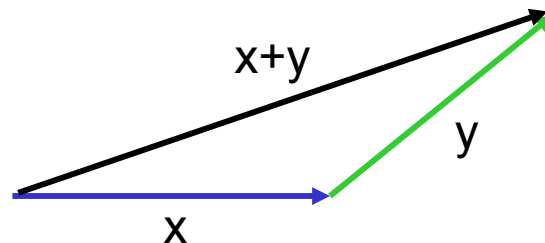
$$\|x\| = 0 \quad \text{si y sólo si } x = (0, \dots, 0) = 0$$

$$\|\alpha x\| = |\alpha| \|x\| \quad \text{para todo } \alpha \in R \text{ y } x \in R^n$$

$$\|x + y\| \leq \|x\| + \|y\| \quad \text{para todo } x, y \in R^n$$

La última propiedad se conoce como desigualdad triangular.

La norma de x se puede pensar como la longitud o magnitud del vector x .



$$\|x + y\| \leq \|x\| + \|y\|$$

Sistemas de Ecuaciones Lineales

Norma de vectores y matrices:

Ejemplo:

La función de R^n en R^+ definida a partir del producto interno de vectores

$$\|x\| = \sqrt{x^t x} = \left(\sum_{i=1}^n x_i^2 \right)^{1/2}$$

es una norma.

Verifiquemos que se cumple la desigualdad triangular.

$$\|x + y\|^2 = (x + y)^t (x + y) = \|x\|^2 + y^t x + x^t y + \|y\|^2$$

Aplicando la desigualdad de Cauchy-Schwarz se tiene

$$\|x + y\|^2 \leq \|x\|^2 + 2\|x\| \|y\| + \|y\|^2 = (\|x\| + \|y\|)^2$$

Tomando raíz cuadrada a ambos lados de la desigualdad sigue

$$\|x + y\| \leq \|x\| + \|y\|$$



Se deja como ejercicio verificar las otras propiedades.

Sistemas de Ecuaciones Lineales

Norma de vectores y matrices (cont.):

Definimos 3 normas vectoriales en R^n

- La norma Euclidea o norma l_2

$$\|x\|_2 = \left(\sum_{i=1}^n x_i^2 \right)^{1/2}$$

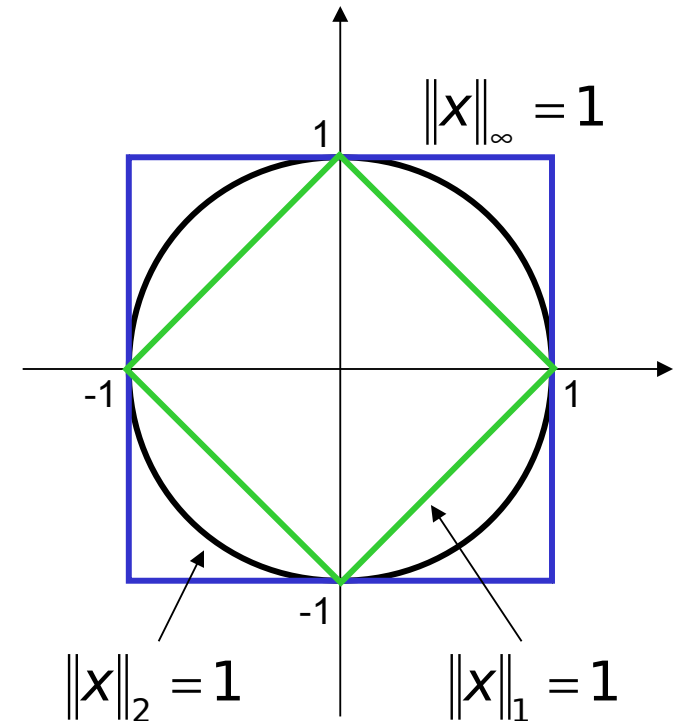
- La norma l_1

$$\|x\|_1 = \sum_{i=1}^n |x_i|$$

- La norma l_∞

$$\|x\|_\infty = \max_{1 \leq i \leq n} |x_i|$$

Ejemplo: conjunto de puntos en R^2 con norma igual a 1



Ejemplo: $x = (1, -1, 3)$

$$\|x\|_1 = |1| + |-1| + |3| = 5, \quad \|x\|_2 = \sqrt{1+1+9} = \sqrt{11}, \quad \|x\|_\infty = \max\{1, 1, 3\} = 3$$

Sistemas de Ecuaciones Lineales

Norma de vectores y matrices (cont.)

- Dos normas vectoriales son equivalentes $\| \cdot \|$ y $\| \cdot \|'$ si existen constantes c_1 y c_2 tales que

$$c_1 \|x\|' \leq \|x\| \leq c_2 \|x\|' \text{ para todo } x \in R^n$$

En la práctica esto significa que cuando $\| \cdot \|'$ está acotada, también $\| \cdot \|$ y viceversa.

Obs. $\|x\|_\infty \leq \|x\|_2 \leq \|x\|_1$ para todo $x \in R^n$

- Una norma matricial es una aplicación

$$\| \cdot \| : M_n \rightarrow R^+ \cup \{0\}$$

que verifica las siguientes propiedades

$$\|A\| = 0 \text{ si y sólo si } A = 0$$

$$\|\lambda A\| = |\lambda| \|A\| \text{ para todo } \lambda \in R, A \in M_n$$

$$\|A + B\| \leq \|A\| + \|B\| \text{ para todo } A, B \in M_n$$

$$\|A \cdot B\| \leq \|A\| \|B\| \text{ para todo } A, B \in M_n$$

Sistemas de Ecuaciones Lineales

Norma de vectores y matrices (cont.)

- Sea $\| \cdot \|$ una norma en R^n , se define la norma matricial

$$\| \cdot \| : M_n \rightarrow R^+ \cup \{0\}$$

como

$$\|A\| = \sup_{x \neq 0} \frac{\|Ax\|}{\|x\|} = \sup_{\|x\|=1} \|Ax\|.$$

Cuando una norma matricial se define de la forma anterior (a través de una norma vectorial), se dice que es una norma matricial subordinada a la norma vectorial. Tenemos los siguiente ejemplos:

$$\|A\|_1 = \sup_{x \neq 0} \frac{\|Ax\|_1}{\|x\|_1}$$

$$\|A\|_2 = \sup_{x \neq 0} \frac{\|Ax\|_2}{\|x\|_2}$$

$$\|A\|_\infty = \sup_{x \neq 0} \frac{\|Ax\|_\infty}{\|x\|_\infty}$$

Sistemas de Ecuaciones Lineales

Norma de vectores y matrices (cont.)

- Algunas propiedades de las normas matriciales subordinadas (demostrarlo)

- * $\|Ax\| \leq \|A\| \|x\|$ para todo $A \in M_n, x \in R^n$

- * Existe un vector $x \in R^n$ para el cual se da la igualdad, es decir,

$$\|Ax\| = \|A\| \|x\|$$

- * Para I la matriz identidad $\|I\| = 1$

- * $\rho(A) \leq \|A\|$ para todo $A \in M_n$

- Normas matriciales subordinadas a las normas vectoriales 1, 2 y ∞

- * $\|A\|_1 = \max_{1 \leq j \leq n} \sum_{i=1}^n |a_{ij}|$

- * $\|A\|_2 = \sqrt{\rho(A^*A)} = \sqrt{\rho(AA^*)} = \|A^*\|_2$

- * $\|A\|_\infty = \max_{1 \leq i \leq n} \sum_{j=1}^n |a_{ij}|$

$\rho(A^*A)$ es el radio espectral de A^*A

Las normas 1 e ∞ se calculan a partir de los elementos de la matriz, la norma 2 no. Es inmediato que $\|A^T\|_\infty = \|A\|_1$.

Sistemas de Ecuaciones Lineales

Norma de vectores y matrices (cont.)

Ejemplo:

$$A = \begin{pmatrix} 1 & 0 & -7 \\ 0 & 2 & 2 \\ -1 & -1 & 0 \end{pmatrix}$$

$$A = [1,0,-7;0,2,2;-1,-1,0]$$

$$\|A\|_1 = \max_{1 \leq j \leq n} \sum_{i=1}^n |a_{ij}| = \max\{2,3,9\} = 9$$

$$\|A\|_\infty = \max_{1 \leq i \leq n} \sum_{j=1}^n |a_{ij}| = \max\{8,4,2\} = 8$$

$$A^T A = \begin{pmatrix} 2 & 1 & -7 \\ 1 & 5 & 4 \\ -7 & 4 & 53 \end{pmatrix} \quad \text{Autovalores: } \begin{cases} 0.5054 \\ 5.2530 \\ 54.2416 \end{cases}$$

$$\|A\|_2 = \sqrt{\rho(A^* A)} = 7.3649$$

MATLAB
norm(A,1)

norm(A,inf)

eig(A' * A)

norm(A,2)

Sistemas de Ecuaciones Lineales

Norma de vectores y matrices (cont.)

- Norma de Frobenius

Es una norma matricial no subordinada a ninguna norma vectorial.

Esta dada por

$$\|A\|_F = \sqrt{\sum_{i,j=1}^n |a_{ij}|^2}$$

Se calcula a partir de los elementos de la matriz.

Ejemplo:

$$A = \begin{pmatrix} 1 & 0 & -7 \\ 0 & 2 & 2 \\ -1 & -1 & 0 \end{pmatrix}$$

MATLAB

`norm(A,'fro')`

$$\|A\|_F = \sqrt{\sum_{i,j=1}^n |a_{ij}|^2} = \sqrt{1 + 49 + 4 + 4 + 1 + 1} = \sqrt{60} = 7.7460$$

Sistemas de Ecuaciones Lineales

El objetivo que perseguimos es resolver numéricamente el sistema de ecuaciones lineales

$$Ax = b$$

donde A es la matriz de los coeficientes del sistema, b es el lado derecho del sistema (o término independiente) y x es el vector de incógnitas o de valores que deseamos hallar.

Si A es una matriz de orden $n \times n$, entonces podemos escribir

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \quad b = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} \quad \text{y} \quad x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$$

Sistemas de Ecuaciones Lineales

Resolución de sistemas de ecuaciones lineales

- **Métodos directos:** proporcionan la solución exacta (salvo errores de redondeo) en un número finito de pasos.
 - Eliminación Gaussiana
 - Sustitución hacia atrás
 - Descomposición LU
 - Sustitución hacia adelante
 - Doolittle, Crout, Cholesky
- **Métodos iterativos:** proporcionan una sucesión $\{x_k\}$ que converge a la solución exacta
 - Richardson
 - Jacobi
 - Gauss Seidel
 - Relajación

Sistemas de Ecuaciones Lineales

Teorema. Para una matriz A de dimensión $n \times n$, las siguientes propiedades son equivalentes:

- i. La inversa de A existe, es decir A es no singular
- ii. El determinante de A es no cero
- iii. Las filas de A forman una base de R^n
- iv. Las columnas de A forman una base de R^n
- v. A como una transformación de R^n en R^n es inyectiva
- vi. A como una transformación de R^n en R^n es sobreyectiva
- vii. La ecuación $Ax = 0$ implica $x = 0$
- viii. Para cada $b \in R^n$, existe un solo $x \in R^n$ tal que $Ax = b$
- ix. A es el producto de matrices elementales
- x. 0 no es un autovalor de A

Sistemas de Ecuaciones Lineales

Sistemas de ecuaciones lineales equivalentes

Definición. Los sistemas

$$Ax = b \quad \text{y} \quad Bx = d$$

cada uno con n ecuaciones y n incógnitas se denominan **sistemas equivalentes** si tienen exactamente la misma solución.

Obs. En muchos casos en lugar de resolver un sistema de ecuaciones lineales, resolveremos un sistema equivalente.

En este caso es importante el no perder o agregar soluciones.

Esta simple idea es el corazón de los procedimientos numéricos.

Sistemas de Ecuaciones Lineales

Operaciones elementales por filas sobre matrices

Están permitidas las operaciones elementales siguientes:

* $\text{fila } i \leftrightarrow \text{fila } j$ intercambio de 2 filas

* $\text{fila } i \leftarrow \lambda \text{ fila } i$ (λ ctte $\neq 0$) multiplicación de una fila por un número distinto de cero

* $\text{fila } i \leftarrow \text{fila } i + \lambda \text{ fila } j$ (λ ctte $\neq 0$) suma de una fila a un múltiplo de otra

Teorema. Si un sistema de ecuaciones

$$Bx = d$$

se obtiene a partir de otro

$$Ax = b$$

mediante una sucesión de operaciones elementales, entonces los 2 sistemas son equivalentes.

Sistemas de Ecuaciones Lineales

Operaciones elementales por filas sobre matrices

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 3 & 3 \\ 2 & 4 & 7 \end{pmatrix}$$

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

matrices
elementales

$$F_2 \leftarrow -1 \cdot F_1 + F_2 \quad E_1 = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$E_1 A = \begin{pmatrix} 1 & 2 & 3 \\ 0 & 1 & 0 \\ 2 & 4 & 7 \end{pmatrix}$$

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

$$F_3 \leftarrow -2 \cdot F_1 + F_3 \quad E_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -2 & 0 & 1 \end{pmatrix}$$

$$E_2 E_1 A = \begin{pmatrix} 1 & 2 & 3 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$E_2 E_1 I = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ -2 & 0 & 1 \end{pmatrix}$$

$$F_1 \leftarrow -2 \cdot F_2 + F_1 \quad E_3 = \begin{pmatrix} 1 & -2 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$E_3 E_2 E_1 A = \begin{pmatrix} 1 & 0 & 3 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$E_3 E_2 E_1 I = \begin{pmatrix} 3 & -2 & 0 \\ -1 & 1 & 0 \\ -2 & 0 & 1 \end{pmatrix}$$

$$F_1 \leftarrow -3 \cdot F_3 + F_1 \quad E_4 = \begin{pmatrix} 1 & 0 & -3 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$E_4 E_3 E_2 E_1 A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$E_4 E_3 E_2 E_1 I = \begin{pmatrix} 9 & -2 & -3 \\ -1 & 1 & 0 \\ -2 & 0 & 1 \end{pmatrix}$$

 matriz inversa de A

Sistemas de Ecuaciones Lineales

Condicionamiento de sistemas

La existencia de un sistema mal condicionado es una fuente de posibles errores y dificultades a la hora de resolver un sistema lineal mediante métodos numéricos.

Problema: definir y cuantificar el condicionamiento de un sistema lineal.

Consideremos el sistema lineal

$$Ax = b \quad (1)$$

con A una matriz $n \times n$ e invertible, b un vector de dimensión n y x la solución exacta del sistema

$$x = A^{-1}b.$$

“pequeños cambios en los datos dan lugar a pequeños cambios en las respuestas”

Sistemas de Ecuaciones Lineales

Condicionamiento de sistemas

- Modifiquemos el vector b mediante una perturbación δb pequeña y nos planteamos resolver el sistema

$$A\hat{x} = b + \delta b.$$

Sea $\delta x = \hat{x} - x$

El sistema esta bien condicionado

si cuando δb es pequeña, δx también lo es. (2)

Observemos

$$A\hat{x} = b + \delta b = Ax + \delta b$$

$$\Leftrightarrow A\hat{x} - Ax = \delta b \quad \Leftrightarrow \quad A(\hat{x} - x) = \delta b \quad \Leftrightarrow \quad A(\delta x) = \delta b \quad (3)$$

es decir, δx es solución del sistema (3), $\delta x = A^{-1}(\delta b)$

Sistemas de Ecuaciones Lineales

Condicionamiento de sistemas (cont.)

Usando la propiedad para las normas matriciales

$$\|\delta x\| = \|A^{-1}(\delta b)\| \leq \|A^{-1}\| \|\delta b\| \quad (4)$$

$$\text{con } x \neq 0 \text{ y } b \neq 0, \quad \|b\| = \|Ax\| \leq \|A\| \|x\| \Leftrightarrow \frac{1}{\|x\|} \leq \|A\| \frac{1}{\|b\|} \quad (5)$$

De (4) y (5)

$$\frac{\|\delta x\|}{\|x\|} \leq \|A^{-1}\| \|A\| \frac{\|\delta b\|}{\|b\|} \quad (6)$$

error relativo vectorial
en los resultados
error relativo vectorial
en los datos

De la relación (6)

parece deducirse que el número $\kappa(A) = \|A^{-1}\| \|A\|$

es el factor determinante en la relación, ya que si este es pequeño tenemos el efecto deseado:

si cuando δb es pequeña, δx también lo es.

Sistemas de Ecuaciones Lineales

Condicionamiento de sistemas (cont.)

- Caso en que las perturbaciones se produzcan en la matriz del sistema (1).
Sea $A + \Delta A$ la matriz perturbada y $x + \Delta x$ la solución aproximada de

$$(A + \Delta A)(x + \Delta x) = b. \quad (6)$$

$$Ax + A\Delta x + (\Delta A)(x + \Delta x) = b \Leftrightarrow A\Delta x + (\Delta A)(x + \Delta x) = 0$$

$$\Leftrightarrow (\Delta A)(x + \Delta x) = -A\Delta x \Leftrightarrow \Delta x = -A^{-1}(\Delta A)(x + \Delta x)$$

chequear

$$\Rightarrow \|\Delta x\| \leq \|A^{-1}\| \|\Delta A\| \|x + \Delta x\| \quad (7)$$

De (7)

error relativo vectorial
en los resultados

$$\frac{\|\Delta x\|}{\|x + \Delta x\|} \leq \|A^{-1}\| \|A\| \frac{\|\Delta A\|}{\|A\|} = \kappa(A) \frac{\|\Delta A\|}{\|A\|}. \quad (8)$$

error relativo
vectorial en los datos

De nuevo el factor determinante en la relación es $\kappa(A)$
ya que si este es pequeño tenemos el efecto deseado.

Sistemas de Ecuaciones Lineales

Condicionamiento de sistemas (cont.)

Definimos el **número de condición** de la matriz A como

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

Obs. El sistema lineal $AX = b$ estará bien condicionado si $\kappa(A)$ es pequeño.
¿cuán pequeño debe ser?

$$1 = \|I\| = \|A^{-1}A\| \leq \|A^{-1}\| \|A\| = \kappa(A) \Leftrightarrow \kappa(A) \geq 1$$

El estará bien condicionado si $\kappa(A)$ se acerca a 1

Propiedades del número de condición de una matriz

- $\kappa(A) \geq 1$,
- $\kappa(A) = \kappa(A^{-1})$,
- $\kappa(\alpha A) = \kappa(A)$ para todo $\alpha \in R - \{0\}$.
- $\kappa(A)$ es infinito si A es singular

Demostración: se deja como ejercicio

Sistemas de Ecuaciones Lineales

Condicionamiento de sistemas (cont.)

Obs. Se tiene la siguiente regla empírica:

Si $\kappa(A) = 10^k$ se espera perder al menos k dígitos al resolver numéricamente el sistema lineal $Ax = b$

Sistemas de Ecuaciones Lineales

Condicionamiento de sistemas (cont.)

Ejemplo:

$$A = \begin{pmatrix} 1 & 0 & -7 \\ 0 & 2 & 2 \\ -1 & -1 & 0 \end{pmatrix} \quad \kappa(A) = \|A^{-1}\| \|A\|$$

$$A^{-1} = \begin{pmatrix} -0.1667 & -0.5833 & -1.1667 \\ 0.1667 & 0.5833 & 0.1667 \\ -0.1667 & -0.0833 & -0.1667 \end{pmatrix}$$

MATLAB

inv(A)

$$\kappa_1(A) = \|A^{-1}\|_1 \|A\|_1 = 13.5$$

cond(A,1)

$$\kappa_2(A) = \|A^{-1}\|_2 \|A\|_2 = 10.3599$$

cond(A,2)

$$\kappa_\infty(A) = \|A^{-1}\|_\infty \|A\|_\infty = 15.3333$$

cond(A,inf)

$$\kappa_F(A) = \|A^{-1}\|_F \|A\|_F = 11.4564$$

cond(A,'fro')

$$\|A\|_1 = \max_{1 \leq j \leq n} \sum_{i=1}^n |a_{ij}|$$

$$\|A\|_\infty = \max_{1 \leq i \leq n} \sum_{j=1}^n |a_{ij}|$$

Sistemas de Ecuaciones Lineales

Condicionamiento de sistemas (cont.)

Ejemplo:

$$A = \begin{pmatrix} 10 & 7 & 8 & 7 \\ 7 & 5 & 6 & 5 \\ 8 & 6 & 10 & 9 \\ 7 & 5 & 9 & 10 \end{pmatrix}, \quad b = \begin{pmatrix} 32 \\ 23 \\ 33 \\ 31 \end{pmatrix}, \quad x = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

$$\tilde{b} = \begin{pmatrix} 32 \\ 22.9 \\ 32.98 \\ 31.02 \end{pmatrix}, \quad \tilde{x} = \begin{pmatrix} 7.28 \\ -9.36 \\ 3.54 \\ -0.5 \end{pmatrix}$$

$$\frac{\|\tilde{b} - b\|}{\|b\|} = \frac{0.4}{119} = 0.0034,$$

$$\frac{\|\tilde{x} - x\|}{\|x\|} = \frac{27.4}{4} = 6.85,$$

$$A^{-1} = \begin{pmatrix} 25 & -41 & 10 & -6 \\ -41 & 68 & -17 & 10 \\ 10 & -17 & 5 & -3 \\ -6 & 10 & -3 & 2 \end{pmatrix}, \quad \kappa_1(A) = 4488$$

¡mal condicionado!

“pequeños
cambios en los
datos dan lugar a
pequeños
cambios en las
respuestas”

Sistemas de Ecuaciones Lineales

Condicionamiento de sistemas (cont.)

Ejemplo: Estudiar el condicionamiento del sistema lineal $Ax = b$ con

$$A = \begin{pmatrix} 1 & 1 + \varepsilon \\ 1 - \varepsilon & 1 \end{pmatrix} \quad \text{con } \varepsilon > 0$$

La inversa $A^{-1} = \frac{1}{\varepsilon^2} \begin{pmatrix} 1 & -\varepsilon - 1 \\ \varepsilon - 1 & 1 \end{pmatrix}$ $\|A\|_{\infty} = \max_{1 \leq i \leq n} \sum_{j=1}^n |a_{ij}|$

$$\|A\|_{\infty} = \max_{1 \leq i \leq n} \sum_{j=1}^n |a_{ij}| = 2 + \varepsilon \quad \|A^{-1}\|_{\infty} = \max_{1 \leq i \leq n} \sum_{j=1}^n |a_{ij}| = \frac{2 + \varepsilon}{\varepsilon^2}$$

$$\kappa_{\infty}(A) = \|A^{-1}\|_{\infty} \|A\|_{\infty} = \frac{(2 + \varepsilon)^2}{\varepsilon^2} > \frac{4}{\varepsilon^2}$$

Si $\varepsilon \leq 0.01$ entonces $\kappa_{\infty}(A) > 40000$

Esto indica que una perturbación de los datos de 0.01 puede originar una perturbación de la solución del sistema de 40000.

Sistemas de Ecuaciones Lineales

Métodos directos

Por medio de las operaciones elementales anteriores, un sistema de ecuaciones lineales

$$Ax = b$$

se puede transformar a un sistema lineal (equivalente) más fácil de resolver y que tiene el mismo conjunto de soluciones.

El principio de los métodos directos que vamos a estudiar reside en determinar una matriz M invertible, tal que la matriz MA sea triangular superior. Tenemos que resolver entonces el sistema lineal

$$MAx = Mb$$

Este principio es la base del **método de Gauss** para la resolución de sistemas lineales con matrices A invertibles.

Sistemas de Ecuaciones Lineales

Método de Gauss.

El método de Gauss es un método general de resolución de un sistema lineal de la forma

$$Ax = b$$

donde A es una matriz invertible.

Esta basado en el siguiente hecho:

si tuviésemos una matriz triangular superior, la resolución numérica del sistema es inmediata.

Este se compone de las siguientes etapas:

- Procedimiento de eliminación, que equivale a determinar una matriz invertible M tal que la matriz MA sea una matriz triangular superior
- Cálculo del vector Mb
- Resolución del sistema lineal $MAx = Mb$, por el método de sustitución hacia atrás.

Sistemas de Ecuaciones Lineales

Matrices de permutación.

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \quad P = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

matriz de permutación

$$PA = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} = \begin{pmatrix} 3 & 4 \\ 1 & 2 \end{pmatrix}$$

intercambia las filas 1 y 2

$$AP = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 2 & 1 \\ 4 & 3 \end{pmatrix}$$

intercambia las columnas 1 y 2

COMPLETAR !!!!!

Sistemas de Ecuaciones Lineales

Método de Gauss. Construcción de MA y Mb

$$A = \begin{pmatrix} 0 & 1 & 2 & 1 \\ 1 & 2 & 1 & 3 \\ 1 & 1 & -1 & 1 \\ 0 & 1 & 8 & 12 \end{pmatrix} \quad b = \begin{pmatrix} 1 \\ 0 \\ 5 \\ 2 \end{pmatrix}$$

matriz
ampliada $\left(\begin{array}{cccc|c} 0 & 1 & 2 & 1 & 1 \\ 1 & 2 & 1 & 3 & 0 \\ 1 & 1 & -1 & 1 & 5 \\ 0 & 1 & 8 & 12 & 2 \end{array} \right)$

$$F_1 \leftrightarrow F_2$$

$$F_3 \leftarrow -1 \cdot F_1 + F_3$$

$$F_3 \leftarrow 1 \cdot F_2 + F_3$$

$$F_4 \leftarrow -1 \cdot F_2 + F_4$$

$$\rightarrow \left(\begin{array}{cccc|c} 1 & 2 & 1 & 3 & 0 \\ 0 & 1 & 2 & 1 & 1 \\ 1 & 1 & -1 & 1 & 5 \\ 0 & 1 & 8 & 12 & 2 \end{array} \right) \rightarrow \left(\begin{array}{cccc|c} 1 & 2 & 1 & 3 & 0 \\ 0 & 1 & 2 & 1 & 1 \\ 0 & -1 & -2 & -2 & 5 \\ 0 & 1 & 8 & 12 & 2 \end{array} \right) \rightarrow \left(\begin{array}{cccc|c} 1 & 2 & 1 & 3 & 0 \\ 0 & 1 & 2 & 1 & 1 \\ 0 & 0 & 0 & -1 & 6 \\ 0 & 0 & 6 & 11 & 1 \end{array} \right)$$

$$F_3 \leftrightarrow F_4$$

$$\rightarrow \left(\begin{array}{cccc|c} 1 & 2 & 1 & 3 & 0 \\ 0 & 1 & 2 & 1 & 1 \\ 0 & 0 & 6 & 11 & 1 \\ 0 & 0 & 0 & -1 & 6 \end{array} \right)$$

matriz triangular superior

$$MA = \begin{pmatrix} 1 & 2 & 1 & 3 \\ 0 & 1 & 2 & 1 \\ 0 & 0 & 6 & 11 \\ 0 & 0 & 0 & -1 \end{pmatrix}, \quad Mb = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 6 \end{pmatrix}$$

Sistemas de Ecuaciones Lineales

Método de Gauss. Construcción de MA y Mb

$$\begin{aligned}
 P_1 &= \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} & E_1 &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} & E_2 &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{pmatrix} & P_3 &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \\
 & F_1 \leftrightarrow F_2 & F_3 \leftarrow -1 \cdot F_1 + F_3 & F_3 \leftarrow 1 \cdot F_2 + F_3 & & F_3 \leftrightarrow F_4 \\
 & & & F_4 \leftarrow -1 \cdot F_2 + F_4 & &
 \end{aligned}$$

$$M = P_3 E_2 E_1 P_1 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 \\ 1 & -1 & 1 & 0 \end{pmatrix}$$

$$\det(M) = \begin{cases} 1 & \text{si } \Lambda \text{ es par} \\ -1 & \text{si } \Lambda \text{ es impar} \end{cases}$$

En la práctica la matriz M no se calcula, sino que se obtiene directamente MA y Mb .

Λ Es el número de matrices de permutación P distintas a la matriz identidad

$$Ax = b \Leftrightarrow MAX = Mb$$

Sistemas de Ecuaciones Lineales

Método de Gauss. Construcción de MA y Mb (otro ejemplo)

$$A = \begin{pmatrix} 6 & -2 & 2 & 4 \\ 12 & -8 & 6 & 10 \\ 3 & -13 & 9 & 3 \\ -6 & 4 & 1 & -18 \end{pmatrix} \quad b = \begin{pmatrix} 16 \\ 26 \\ -19 \\ -34 \end{pmatrix}$$

$$A = [6 \ -2 \ 2 \ 4; 12 \ -8 \ 6 \ 10; 3 \ -13 \ 9 \ 3; -6 \ 4 \ 1 \ -18]$$

$$b = [16; 26; -19; -34]$$

operaciones elementales sucesivas:

$$\left. \begin{array}{l} F_2 \leftarrow -2.F_1 + F_2 \\ F_3 \leftarrow -\frac{1}{2}.F_1 + F_3 \\ F_4 \leftarrow -(-1).F_1 + F_4 \end{array} \right\} \longrightarrow A = \begin{pmatrix} 6 & -2 & 2 & 4 \\ 0 & -4 & 2 & 2 \\ 0 & -12 & 8 & 1 \\ 0 & 2 & 3 & -14 \end{pmatrix}, \quad b = \begin{pmatrix} 16 \\ -6 \\ -27 \\ -18 \end{pmatrix}$$

Sistemas de Ecuaciones Lineales

Método de Gauss. Construcción de MA y Mb (otro ejemplo) (cont.)

$$\left. \begin{array}{l} F_3 \leftarrow -3.F_2 + F_3 \\ F_4 \leftarrow -\left(-\frac{1}{2}\right)F_2 + F_4 \end{array} \right\} \longrightarrow A = \begin{pmatrix} 6 & -2 & 2 & 4 \\ 0 & -4 & 2 & 2 \\ 0 & 0 & 2 & -5 \\ 0 & 0 & 4 & -13 \end{pmatrix}, \quad b = \begin{pmatrix} 16 \\ -6 \\ -9 \\ -21 \end{pmatrix}$$

$$\left. \begin{array}{l} F_4 \leftarrow -2.F_3 + F_4 \end{array} \right\} \longrightarrow A = \begin{pmatrix} 6 & -2 & 2 & 4 \\ 0 & -4 & 2 & 2 \\ 0 & 0 & 2 & -5 \\ 0 & 0 & 0 & -3 \end{pmatrix}, \quad b = \begin{pmatrix} 16 \\ -6 \\ -9 \\ -3 \end{pmatrix}$$

sistema triangular superior: $MA = \begin{pmatrix} 6 & -2 & 2 & 4 \\ 0 & -4 & 2 & 2 \\ 0 & 0 & 2 & -5 \\ 0 & 0 & 0 & -3 \end{pmatrix}, \quad Mb = \begin{pmatrix} 16 \\ -6 \\ -9 \\ -3 \end{pmatrix}$

Sistemas de Ecuaciones Lineales

Método de Gauss. Construcción de MA y Mb (otro ejemplo) (cont.)

M es el producto de matrices elementales:

$$\left. \begin{array}{l} F_2 \leftarrow -2.F_1 + F_2 \\ F_3 \leftarrow -\frac{1}{2}.F_1 + F_3 \\ F_4 \leftarrow -(-1).F_1 + F_4 \end{array} \right\} \longrightarrow$$

$$E_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -2 & 1 & 0 & 0 \\ -1/2 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}$$

$$E_1 = [1 \ 0 \ 0 \ 0; -2 \ 1 \ 0 \ 0; -1/2 \ 0 \ 1 \ 0; 1 \ 0 \ 0 \ 1]$$

$$\left. \begin{array}{l} F_3 \leftarrow -3.F_2 + F_3 \\ F_4 \leftarrow -\left(-\frac{1}{2}\right)F_2 + F_4 \end{array} \right\} \longrightarrow$$

$$E_2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -3 & 1 & 0 \\ 0 & 1/2 & 0 & 1 \end{pmatrix}$$

$$E_2 = [1 \ 0 \ 0 \ 0; 0 \ 1 \ 0 \ 0; 0 \ -3 \ 1 \ 0; 0 \ 1/2 \ 0 \ 1]$$

$$F_4 \leftarrow -2.F_3 + F_4 \left. \right\} \longrightarrow$$

$$E_3 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -2 & 1 \end{pmatrix}$$

$$E_3 = [1 \ 0 \ 0 \ 0; 0 \ 1 \ 0 \ 0; 0 \ 0 \ 1 \ 0; 0 \ 0 \ -2 \ 1]$$

Sistemas de Ecuaciones Lineales

Método de Gauss. Construcción de MA y Mb (otro ejemplo) (cont.)

M es el producto de matrices elementales:

$$M = E_3 E_2 E_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -2 & 1 & 0 & 0 \\ 11/2 & -3 & 1 & 0 \\ -11 & 13/2 & -2 & 1 \end{pmatrix}$$

y se cumple que:

$$MA = \begin{pmatrix} 6 & -2 & 2 & 4 \\ 0 & -4 & 2 & 2 \\ 0 & 0 & 2 & -5 \\ 0 & 0 & 0 & -3 \end{pmatrix} \quad y \quad Mb = \begin{pmatrix} 16 \\ -6 \\ -9 \\ -3 \end{pmatrix}$$

Obs. Notar que la inversa de una matriz elemental E_1 y el producto de las matrices elementales E_1^{-1} , E_2^{-1} y E_3^{-1} es muy sencillo de construir

$$E_1^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 1/2 & 0 & 1 & 0 \\ -1 & 0 & 0 & 1 \end{pmatrix} \quad y \quad E_1^{-1} E_2^{-1} E_3^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 1/2 & 3 & 1 & 0 \\ -1 & -1/2 & 2 & 1 \end{pmatrix}$$

Sistemas de Ecuaciones Lineales

Método de Gauss. Ejemplo:

$$\left. \begin{array}{l} x + 2y + z = 3 \\ x + y + 2z = 9 \\ 2x + y + z = 16 \end{array} \right\} \Rightarrow \left. \begin{array}{l} x + 2y + z = 3 \\ -y + z = 6 \\ -3y - z = 10 \end{array} \right\} \Rightarrow \left. \begin{array}{l} x + 2y + z = 3 \\ -y + z = 6 \\ -4z = -8 \end{array} \right\} \Rightarrow \begin{array}{l} z = 2 \\ y = -4 \\ x = 9 \end{array}$$

$$F_2 \leftarrow -1 \cdot F_1 + F_2$$

$$F_3 \leftarrow -2 \cdot F_1 + F_3$$

$$F_3 \leftarrow -3 \cdot F_2 + F_3$$

$$\left. \left[\begin{array}{ccc} 1 & & \\ & 1 & \\ & 3 & 1 \end{array} \right]^{-1} \left[\begin{array}{ccc} 1 & & \\ 1 & 1 & \\ 2 & & 1 \end{array} \right]^{-1} \left[\begin{array}{ccc} 1 & 2 & 1 \\ 1 & 1 & 2 \\ 2 & 1 & 1 \end{array} \right] = \left[\begin{array}{ccc} 1 & 2 & 1 \\ & -1 & 1 \\ & & -4 \end{array} \right] \right\} \Rightarrow \left[\begin{array}{ccc} 1 & 2 & 1 \\ 1 & 1 & 2 \\ 2 & 1 & 1 \end{array} \right] \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 3 \\ 9 \\ 6 \end{bmatrix}$$

$$\left. \left[\begin{array}{ccc} 1 & 2 & 1 \\ 1 & 1 & 2 \\ 2 & 1 & 1 \end{array} \right] = \left[\begin{array}{ccc} 1 & & \\ 1 & 1 & \\ 2 & 3 & 1 \end{array} \right] \left[\begin{array}{ccc} 1 & 2 & 1 \\ & -1 & 1 \\ & & -4 \end{array} \right] \right\} \Rightarrow \left[\begin{array}{ccc} 1 & & \\ 1 & 1 & \\ 2 & 3 & 1 \end{array} \right] \left[\begin{array}{ccc} 1 & 2 & 1 \\ & -1 & 1 \\ & & -4 \end{array} \right] \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 3 \\ 9 \\ 6 \end{bmatrix}$$

Sistemas de Ecuaciones Lineales

Algoritmo de eliminación en el método de Gauss.

Vamos a suponer que los elementos de la diagonal (pivotes) de las matrices sucesivas que surgen durante este método, son no nulos.

Leer A y b

Para $k = 2$ hasta n \longrightarrow $k-1$ fila pivote

Para $i = k$ hasta n \longrightarrow modifica filas k a la n

$$\alpha = a_{i,k-1} / a_{k-1,k-1}$$

Para $j = k$ hasta n \longrightarrow modifica columnas k a la n

$$a_{i,j} = a_{i,j} - \alpha a_{k-1,j}$$

Fin para

$b_i = b_i - \alpha b_{k-1}$ \longrightarrow modifica el lado derecho

Fin para

Fin para

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

Obs. Al finalizar sabemos que U (matriz triangular superior MA) va a estar almacenada en la parte triangular superior de la matriz A , incluyendo la diagonal, y que el término de la derecha (es decir, Mb) está almacenado en el mismo vector b

Sistemas de Ecuaciones Lineales

Algoritmo de eliminación en el método de Gauss.

En general algunos de los elementos de la diagonal de las matrices sucesivas, pueden ser cero. Esto se solventa permutando la filas en la matriz A , lo cual corresponde a multiplicar A por una matriz elemental de permutación P .

Leer A y b

Para $k = 2$ hasta n

Buscar un índice i_0 tal que $(k-1 \leq i_0 \leq n)$ **y** $(a_{i_0, k-1} \neq 0)$

Permutar la fila $k-1$ con la fila i_0

Para $i = k$ hasta n

$$\alpha = a_{i, k-1} / a_{k-1, k-1}$$

Para $j = k$ hasta n

$$a_{i, j} = a_{i, j} - \alpha a_{k-1, j}$$

Fin para

$$b_i = b_i - \alpha b_{k-1}$$

Fin para

Fin para

estrategia de pivoteo

Obs. La mejor manera de escoger i_0 es tal que

$$|a_{i_0, k-1}| \geq |a_{i, k-1}| \quad \text{para } k-1 \leq i \leq n$$

Sistemas de Ecuaciones Lineales

Algoritmo de eliminación en el método de Gauss con pivoteo

Leer A y b en una matriz $C = (A | b)$

Para $k = 2$ hasta n

$i_0 = k-1$

Para $i=k$ hasta n

Si $|C_{i_0,k-1}| < |C_{i,k-1}|$

$i_0 = i$

Fin si

Fin para

Para $j=1$ hasta $n+1$

$S = C_{i_0,j}; C_{i_0,j} = C_{k-1,j}; C_{k-1,j} = S$

Fin para

Para $i = k$ hasta n

$\alpha = C_{i,k-1} / C_{k-1,k-1}$

Para $j = k$ hasta $n+1$

$C_{i,j} = C_{i,j} - \alpha C_{k-1,j}$

Fin para

Fin para

Fin para

$$C = \begin{pmatrix} C_{11} & C_{12} & \cdots & C_{1n} & C_{1,n+1} \\ C_{21} & C_{22} & \cdots & C_{2n} & C_{2,n+1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ C_{n1} & C_{n2} & \cdots & C_{nn} & C_{n,n+1} \end{pmatrix}$$

escoger el mejor i_0

permutar la fila $k-1$ con i_0

modificar elementos de
las filas k a la n y
columnas k a la $n+1$

Sistemas de Ecuaciones Lineales

Método de sustitución hacia atrás.

$$MAx = Mb \quad \text{donde} \quad MA = \begin{pmatrix} 1 & 2 & 1 & 3 \\ 0 & 1 & 2 & 1 \\ 0 & 0 & 6 & 11 \\ 0 & 0 & 0 & -1 \end{pmatrix} \quad Mb = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 6 \end{pmatrix}$$

$$\det(MA) = 1 \cdot 1 \cdot 6 \cdot (-1) = -6 \neq 0$$

En forma de ecuaciones

$$\begin{cases} x_1 + 2x_2 + x_3 + 3x_4 = 0 \\ x_2 + 2x_3 + x_4 = 1 \\ 6x_3 + 11x_4 = 1 \\ -x_4 = 6 \end{cases} \Leftrightarrow \begin{cases} x_1 = -2x_2 - x_3 - 3x_4 \\ x_2 = 1 - 2x_3 - x_4 \\ x_3 = (1 - 11x_4) / 6 \\ x_4 = -6 \end{cases} \Leftrightarrow x = \begin{pmatrix} 37.5 \\ -15.3333 \\ 11.1667 \\ -6 \end{pmatrix}$$

Algoritmo general
para resolver el
sistema $Ux = d$

$$x_n = d_n / u_{nn}$$

$$x_i = (d_i - \sum_{j=i+1}^n u_{ij}x_j) / u_{ii}$$

para $i = n - 1$ hasta 1

Sistemas de Ecuaciones Lineales

Método de sustitución hacia atrás.

Resolución de un sistema de ecuaciones lineales triangular superior

```

1  %Resuelve el sistema triangular superior Ux=b usando sustitucion hacia atras
2  %Entradas: U = matriz cuadrada triangular superior,
3  %          b = vector de la misma dimension
4  %Salida:   x = vector (solucion)
5
6  U = input('matriz ');
7  %disp(U);
8  b = input('vector ');
9  %disp(b);
10 [n m]=size(U);
11 l = length(b);
12
13 if (n == m & n == l)
14     for i = n:-1:1
15         sum =0.;
16         for j = i+1:n
17             sum = sum + U(i,j)*x(j);
18         end
19         x(i) = (b(i) - sum)/U(i,i);
20     end
21     disp('vector solucion')
22     disp(x);
23 else
24     disp('matriz no es cuadrada o');
25     disp('las dimensiones de la matriz y el vector columna no son compatibles');
26 end

```

U una matriz $n \times n$
triangular superior,
 b vector de longitud n .
Resolver el sistema

$$Ux = b$$

$$U = [1,2,1,3; 0,1,2,1; 0,0,6,11; 0,0,0,-1]$$

$$b = [0,1,1,6]$$

Algoritmo

$$x_n = b_n / u_{nn}$$

$$x_i = (b_i - \sum_{j=i+1}^n u_{ij} x_j) / u_{ii}$$

para $i = n-1$ hasta 1

Sistemas de Ecuaciones Lineales

Estrategias de pivoteo - Ejemplo

$$\begin{pmatrix} 0.003 & 59.14 \\ 5.291 & -6.13 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 59.17 \\ 46.78 \end{pmatrix} \quad \text{solución} \quad \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 10 \\ 1 \end{pmatrix}$$

Aplicamos el método de Gauss usando aritmética de 4 dígitos

$$\left(\begin{array}{cc|c} 0.003 & 59.14 & 59.17 \\ 5.291 & -6.13 & 46.78 \end{array} \right) \quad \alpha = -\frac{5.291}{0.003} = -1763.66 \quad \rightarrow \quad \alpha = -1764$$

$$\left(\begin{array}{cc|c} 0.003 & 59.14 & 59.17 \\ 0 & -104300 & -104400 \end{array} \right) \quad \text{usando sustitución hacia atrás}$$

$$x_1 = \frac{59.17 - 59.14 \cdot 1.001}{0.003} = -10.0 \quad \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} -10 \\ 1.001 \end{pmatrix}$$

El error es grande en x_1 , aunque es pequeño de 0.001 en x_2 .

Este ejemplo ilustra las dificultades que pueden surgir en algunos casos cuando el elemento pivote, es pequeño en relación a los elementos de columna en la matriz.

Sistemas de Ecuaciones Lineales

Estrategias de pivoteo – Ejemplo (cont.)

$$\begin{pmatrix} 0.003 & 59.14 \\ 5.291 & -6.13 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 59.17 \\ 46.78 \end{pmatrix} \quad \text{solución} \quad \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 10 \\ 1 \end{pmatrix}$$

La idea es seleccionar el elemento en la misma columna que está debajo de la diagonal y que tiene el mayor valor absoluto.

Permutamos las filas F_1 y F_2 , ya que el pivote (0.003) es pequeño respecto a los otros elementos en la columna

$$\left(\begin{array}{cc|c} 5.291 & -6.13 & 46.78 \\ 0.003 & 59.14 & 59.17 \end{array} \right) \quad \alpha = -\frac{0.003}{5.291} = -0.000567$$

$$\left(\begin{array}{cc|c} 5.291 & -6.13 & 46.78 \\ 0 & 59.14 & 59.14 \end{array} \right) \quad \text{usando sustitución hacia atrás}$$

$$x_1 = \frac{46.78 + 6.13 \cdot 1}{5.291} = 10.0 \quad \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 10 \\ 1 \end{pmatrix}$$

Sistemas de Ecuaciones Lineales

Conteo de operaciones de punto flotante (+ , - , * , /) para el método de Gauss

- Triangularización del sistema

Número de sumas y restas: $nop1$

para $k=2$, $n \cdot (n-1)$

para $k=3$, $(n-1) \cdot (n-2)$

...

para $k=n$, $2 \cdot 1$

$$\left. \begin{array}{l} \text{para } k=2, n \cdot (n-1) \\ \text{para } k=3, (n-1) \cdot (n-2) \\ \dots \\ \text{para } k=n, 2 \cdot 1 \end{array} \right\} \sum_{k=2}^n (n-k+2)(n-k+1)$$

Obs.

$$\sum_{k=1}^m k = \frac{m(m+1)}{2}$$

$$\sum_{k=1}^m k^2 = \frac{m(m+1)(2m+1)}{6}$$

$$nop1 = \sum_{k=2}^n (n-k+2)(n-k+1) = \sum_{k=1}^{n-1} (n-k+1)(n-k)$$

$$= \sum_{k=1}^{n-1} n^2 + (-2k+1)n + k^2 - k$$

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

$$= n(n-1) \left[1 + \frac{1}{3}n - \frac{1}{6} - \frac{1}{2} \right] = \frac{1}{3}n(n-1)(n+1)$$

$$nop1 = \frac{n^3 - n}{3} \quad (9)$$

Sistemas de Ecuaciones Lineales

Conteo de operaciones de punto flotante para el método de Gauss (cont.)

Número de multiplicaciones y divisiones: $nop2$

Número de multiplicaciones = $nop1$

Número de divisiones

$$\left. \begin{array}{l} \text{para } k=2, (n-1) \\ \text{para } k=3, (n-2) \\ \dots \\ \text{para } k=n, 1 \end{array} \right\} \sum_{k=2}^n (n - k + 1)$$

$$nop2 = nop1 + \sum_{k=2}^n (n - k + 1) = nop1 + \sum_{k=1}^{n-1} (n - k)$$

$$= nop1 + n(n-1) - \frac{n(n-1)}{2} = \frac{n^3 - n}{3} + \frac{n^2 - n}{2} = \frac{n^3}{3} + \frac{n^2}{2} - \frac{5n}{6}$$

$$nop2 = \frac{n^3}{3} + \frac{n^2}{2} - \frac{5n}{6} \quad (10)$$

De (9) y (10) el número total de operaciones: nop

$$nop = nop1 + nop2 = \frac{n^3 - n}{3} + \frac{n^3}{3} + \frac{n^2}{2} - \frac{5n}{6} = \frac{2n^3}{3} + \frac{n^2}{2} - \frac{7n}{6} \quad (11)$$

Sistemas de Ecuaciones Lineales

Conteo de operaciones de punto flotante para el método de Gauss (cont.)

- Resolución del sistema triangular

Número de sumas y restas: $nop3$

$$nop3 = \sum_{i=n-1}^1 (n - i + 1) = \sum_{i=1}^{n-1} (i + 1) = \frac{(n-1)n}{2} + n - 1 = \frac{n^2}{2} + \frac{n}{2} - 1 \quad (12)$$

Número de multiplicaciones y divisiones: $nop4$

$$nop4 = \left[\sum_{i=n-1}^1 (n - i + 1) \right] + 1 = \left[\sum_{i=1}^{n-1} (i + 1) \right] + 1 = \frac{n^2}{2} + \frac{n}{2} \quad (13)$$

De (12) y (13) el número total de operaciones: nop

$$nop = \frac{n^2}{2} + \frac{n}{2} - 1 + \frac{n^2}{2} + \frac{n}{2} = \boxed{n^2 + n - 1} \quad (14)$$

Finalmente, de (11) y (14), el método de Gauss involucra el número total de operaciones

$$\frac{2n^3}{3} + \frac{n^2}{2} - \frac{7n}{6} + n^2 + n - 1 = \boxed{\frac{2n^3}{3} + \frac{3n^2}{2} - \frac{n}{6} - 1} \rightarrow O(n^3)$$

Sistemas de Ecuaciones Lineales

Conteo de operaciones de punto flotante (+ , - , * , /)

Observaciones:

A manera de ilustración, supongamos que el sistema lineal a ser resuelto tiene dimensión 5 mil, es decir $n = 5000$. Entonces, la cantidad de operaciones de punto flotante que se requerirán para obtener una aproximación numérica de la solución de dicho sistema, es aproximadamente $n^3 = 125 \times 10^9$. Si se dispone de un computador con una capacidad de procesamiento de 200 mega flops (200 millones de operaciones de punto flotante por segundo), entonces se requerirán

$125 \times 10^9 / 200 \times 10^6 = 625$ segundos aprox. para obtener la solución. Esto equivale a un poco más de 10 minutos, lo cual es un tiempo considerable, sobre todo si se deben resolver varios sistemas de tamaño similar. Cabe destacar que este valor de n no es demasiado grande; de hecho, los problemas reales clásicos manejan sistemas lineales del orden de cientos de miles

Sistemas de Ecuaciones Lineales

Método de descomposición LU.

Este consiste en encontrar una descomposición de la matriz cuadrada A invertible de la forma

$$A = LU$$

donde L es una matriz triangular inferior y U es una matriz triangular superior.

Supondremos que no es necesario realizar permutaciones de las filas de A , y procedemos como en el método de Gauss cuando pasamos del sistema lineal $Ax=b$ al sistema triangular superior $Ux=d$, con

$$U = E_{n-1} \cdots E_1 A$$

donde E_i es una matriz elemental, para $i=1, \dots, n-1$.

$$E_i = \begin{pmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & e_{i+1,i} & 1 & & \\ & & \vdots & & \ddots & \\ & & e_{n,i} & & & 1 \end{pmatrix} \xrightarrow{\text{inversa}} E_i^{-1} = \begin{pmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & -e_{i+1,i} & 1 & & \\ & & \vdots & & \ddots & \\ & & -e_{n,i} & & & 1 \end{pmatrix}$$

Sistemas de Ecuaciones Lineales

Método de descomposición LU (cont.)

Algoritmo para calcular L y U simultáneamente.

Al final del proceso U está almacenada en la parte triangular superior de A y L en la parte triangular inferior estricta de A (no se incluye la diagonal)

Leer A

Para $k = 2$ hasta n

Para $i = k$ hasta n

$$a_{i,k-1} = a_{i,k-1} / a_{k-1,k-1}$$

Para $j = k$ hasta n

$$a_{i,j} = a_{i,j} - a_{i,k-1} a_{k-1,j}$$

Fin para

Fin para

Fin para

cálculo de L

cálculo de U

Ejemplo: egs_LU.m

Sistemas de Ecuaciones Lineales

Método de descomposición LU (cont.)

Luego para resolver el sistema $AX = b$ debemos resolver $LUX = b$ el cual lo operamos en 2 pasos (realizando el cambio $y = UX$)

$$* Ly = b$$

$$* UX = y$$

Algoritmo para resolver
 $LUX = b$

$$\left. \begin{array}{l} x_1 = b_1 \\ \text{Para } i = 2 \text{ hasta } n \\ \quad x_i = b_i - \sum_{j=1}^{i-1} a_{ij} x_j \\ \text{Fin para} \\ x_n = x_n / a_{nn} \\ \text{Para } i = n-1 \text{ hasta } 1 \\ \quad x_i = \left(x_i - \sum_{j=i+1}^n a_{ij} x_j \right) / a_{ii} \\ \text{Fin para} \end{array} \right\} \begin{array}{l} \text{Resolver} \\ Ly=b \\ \\ \\ \text{Resolver} \\ Ux=y \end{array}$$

Obs. En este algoritmo se está usando la matriz $A = (a_{ij})$ que sale del método de descomposición LU basado en Gauss, mostrado en la lamina anterior.

Ejercicio: Calcular el número de operaciones elementales para resolver un sistema lineal de ecuaciones usando el método de descomposición LU .

Sistemas de Ecuaciones Lineales

Método de descomposición LU (cont.)

Condición necesaria y suficiente para que una matriz admita una descomposición LU .

Teorema:

Sea $A = (a_{ij})_{1 \leq i, j \leq n}$ una matriz invertible. Denotemos por A_{pp} la submatriz de A siguiente

$$A_{pp} = (a_{ij})_{1 \leq i, j \leq p}, \quad 1 \leq p \leq n$$

El $\det(A_{pp}) \neq 0$, $1 \leq p \leq n$ si y sólo si la matriz A admite una descomposición LU .

Ejercicio. Usando el teorema anterior, ¿qué puede decir de las matrices siguientes?

$$A = \begin{pmatrix} 0. & 2. \\ 3. & 1. \end{pmatrix} \quad A = \begin{pmatrix} 0. & 1. \\ 0. & 2. \end{pmatrix}$$

Sistemas de Ecuaciones Lineales

Método de descomposición LU (cont.)

$$A = \begin{pmatrix} 0. & 2. \\ 3. & 1. \end{pmatrix}$$

A es invertible, $\det(A) = -6$.

$$\det(A_{11}) = 0, \quad \det(A_{22}) = -6$$

descomposición LU para A

$$L = \begin{pmatrix} 1. & 0. \\ 0. & 1. \end{pmatrix}, \quad U = \begin{pmatrix} 3. & 1. \\ 0. & 2. \end{pmatrix}, \quad P = \begin{pmatrix} 0. & 1. \\ 1. & 0. \end{pmatrix}$$

$$A = PLU$$

$$A = \begin{pmatrix} 0. & 1. \\ 0. & 2. \end{pmatrix}$$

A no es invertible, $\det(A) = 0$.

$$\det(A_{11}) = 0, \quad \det(A_{22}) = 0$$

descomposición LU para A

$$L = \begin{pmatrix} 1. & 0. \\ 0. & 1. \end{pmatrix}, \quad U = \begin{pmatrix} 0. & 1. \\ 0. & 2. \end{pmatrix}$$

$$A = LU$$

Sistemas de Ecuaciones Lineales

Definición:

Una matriz A de orden $n \times n$ es **diagonal dominante estricta** si

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

Consecuencia:

1. Si A es diagonal dominante estricta entonces todas las matrices equivalentes obtenidas en el método de Gauss, también son diagonalmente estrictas. En este caso no se requiere pivoteo para el método de Gauss.
2. Si A es diagonal dominante estricta entonces A es invertible.
3. Si A es diagonal dominante estricta entonces A admite una descomposición LU .

Definición:

Una matriz A de orden $n \times n$ es **definida positiva** si

$$x^T A x > 0 \quad \text{para todo } x \in R^n - \{0\}$$

Consecuencia:

- Si A es simétrica, entonces A es definida positiva si y sólo si existe una matriz L invertible triangular inferior tal que $A = L L^T$

Sistemas de Ecuaciones Lineales

Ejemplo:

La matriz

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

es diagonal dominante estricta, ya que $|a_{ii}| > \sum_{j=1, j \neq i}^n |a_{ij}|$, $1 \leq i \leq n$

$$|7| > |2| + |0|$$

$$|5| > |3| + |-1|$$

$$|-6| > |0| + |5|$$

Es interesante notar sin embargo que A^T no es diagonal dominante estricta

$$A^T = \begin{pmatrix} 7 & 3 & 0 \\ 2 & 5 & 5 \\ 0 & -1 & -6 \end{pmatrix}$$

$$|7| > |3| + |0|$$

$$|5| < |2| + |5|$$

$$|-6| > |0| + |-1|$$

Sistemas de Ecuaciones Lineales

Ejemplo:

La matriz

$$A = \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix}$$

es positiva definida, ya que $x^T Ax > 0$ para todo $x \in R^n - \{0\}$

$$\begin{aligned} x^T Ax &= (x_1 \quad x_2 \quad x_3) \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = (x_1 \quad x_2 \quad x_3) \begin{pmatrix} 2x_1 - x_2 \\ -x_1 + 2x_2 - x_3 \\ -x_2 + 2x_3 \end{pmatrix} \\ &= 2x_1^2 - 2x_1x_2 + 2x_2^2 - 2x_2x_3 + 2x_3^2 \\ &= x_1^2 + (x_1 - x_2)^2 + (x_2 - x_3)^2 + x_3^2 \end{aligned}$$

Finalmente $x^T Ax = x_1^2 + (x_1 - x_2)^2 + (x_2 - x_3)^2 + x_3^2 > 0$

a menos que $x_1 = x_2 = x_3 = 0$

Sistemas de Ecuaciones Lineales

Teorema:

Sea $A = (a_{ij})_{1 \leq i, j \leq n}$ una matriz simétrica. Denotemos por A_{pp} la submatriz de A siguiente

$$A_{pp} = (a_{ij})_{1 \leq i, j \leq p}, \quad 1 \leq p \leq n$$

El $\det(A_{pp}) > 0$, $1 \leq p \leq n$ si y sólo si la matriz A es definida positiva.

Ejemplo: En el ejemplo anterior se uso la definición para demostrar que la matriz simétrica A es definida positiva. Para confirmar esto usando el resultado de arriba notar que

$$A = \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix}$$

$$\det(A_{11}) = 2 > 0 \quad \det(A_{22}) = \begin{vmatrix} 2 & -1 \\ -1 & 2 \end{vmatrix} = 3 > 0$$

$$\det(A_{22}) = \begin{vmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{vmatrix} = 4 > 0$$

Sistemas de Ecuaciones Lineales

Escalamiento de una matriz

Supongamos que se quiere resolver el sistema lineal con matriz ampliada (usando aritmética de 4 dígitos)

$$A_1 = \left(\begin{array}{cc|c} 2 \cdot 10^{-5} & 1 & 1 \\ 1 \cdot 10^{-5} & 1 \cdot 10^{-5} & 2 \cdot 10^{-5} \end{array} \right) \quad \text{con solución} \quad x = \begin{pmatrix} 1.00002 \\ 0.99998 \end{pmatrix}$$

- Aplicando el método de Gauss sin y con pivoteo

$$\left(\begin{array}{cc|c} 2 \cdot 10^{-5} & 1 & 1 \\ 0 & -0.5 & -0.5 \end{array} \right) \Rightarrow x = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad \text{error del 100\%}$$

chequear

Sistemas de Ecuaciones Lineales

Escalamiento de una matriz (cont.)

- Multiplicando la F_2 de A_1 por 10^5

$$A_1 = \left(\begin{array}{cc|c} 2 \cdot 10^{-5} & 1 & 1 \\ 1 \cdot 10^{-5} & 1 \cdot 10^{-5} & 2 \cdot 10^{-5} \end{array} \right)$$

$$A_2 = \left(\begin{array}{cc|c} 2 \cdot 10^{-5} & 1 & 1 \\ 1 & 1 & 2 \end{array} \right)$$

aplicando el método de Gauss
con y sin pivoteo

Gauss sin pivoteo $\left(\begin{array}{cc|c} 2 \cdot 10^{-5} & 1 & 1 \\ 0 & -0.5 \cdot 10^5 & -0.5 \cdot 10^5 \end{array} \right) \Rightarrow x = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ error del 100%

Gauss con pivoteo $\left(\begin{array}{cc|c} 1 & 1 & 2 \\ 0 & 1 & 1 \end{array} \right) \Rightarrow x = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ solución aceptable

Sistemas de Ecuaciones Lineales

Escalamiento de una matriz (cont.)

- Multiplicando todo el sistema por 10^5 se obtiene

$$A_1 = \left(\begin{array}{cc|c} 2 \cdot 10^{-5} & 1 & 1 \\ 1 \cdot 10^{-5} & 1 \cdot 10^{-5} & 2 \cdot 10^{-5} \end{array} \right)$$

$$A_3 = \left(\begin{array}{cc|c} 2 & 1 \cdot 10^5 & 1 \cdot 10^5 \\ 1 & 1 & 2 \end{array} \right) \quad \text{aplicando el método de Gauss con y sin pivoteo}$$

$$\left(\begin{array}{cc|c} 2 & 10^5 & 10^5 \\ 0 & -0.5 \cdot 10^5 & -0.5 \cdot 10^5 \end{array} \right) \Rightarrow x = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad \text{error del 100\%}$$

Sistemas de Ecuaciones Lineales

Escalamiento de una matriz (cont.)

Obs. La diferencia esencial entre A_1 , A_2 y A_3 es que

$$\begin{aligned} |\det(A_1)| &\approx 10^{-5} \\ |\det(A_2)| &\approx 1 \\ |\det(A_3)| &\approx 10^5 \end{aligned} \quad \text{donde}$$

$$A_1 = \left(\begin{array}{cc|c} 2 \cdot 10^{-5} & 1 & 1 \\ 1 \cdot 10^{-5} & 1 \cdot 10^{-5} & 2 \cdot 10^{-5} \end{array} \right)$$

$$A_2 = \left(\begin{array}{cc|c} 2 \cdot 10^{-5} & 1 & 1 \\ & 1 & 2 \end{array} \right)$$

$$A_3 = \left(\begin{array}{cc|c} 2 & 1 \cdot 10^5 & 1 \cdot 10^5 \\ 1 & 1 & 2 \end{array} \right)$$

Para resolver el sistema $Ax = b$, el **método de escalamiento** consiste en reemplazar la matriz A por la matriz D_1AD_2 , con D_1 y D_2 matrices diagonales invertibles para obtener el sistema equivalente

$$D_1AD_2(D_2^{-1}x) = D_1b \quad (*)$$

con $|\det(D_1AD_2)| \approx 1$

El sistema (*) es equivalente al sistema

$$\begin{cases} D_1AD_2y = D_1b \\ x = (D_2y) \end{cases}$$

Sistemas de Ecuaciones Lineales

Escalamiento de una matriz (cont.)

Obs.

- La escogencia de D_1 y D_2 matrices diagonales corresponde únicamente a una razón de facilidad en los cálculos
- Usualmente se utilizan las cantidades

$$d_{1_i} = \left(\max_{1 \leq j \leq n} |a_{ij}| \right)^{-1} \quad \text{i-ésimo elemento de la diagonal de la matriz } D_1$$

(máximo sobre la fila i)

$$d_{2_i} = \left(\max_{1 \leq j \leq n} |a_{ji}| \right)^{-1} \quad \text{i-ésimo elemento de la diagonal de la matriz } D_2$$

(máximo sobre la columna i)

- Normalmente se utiliza escalamiento por filas, es decir se toma D_2 igual a la matriz identidad

Sistemas de Ecuaciones Lineales

Escalamiento de una matriz (cont.)

Ejemplo. Queremos resolver el sistema $Ax = b$ con

$$A = \begin{pmatrix} 2 \cdot 10^{-5} & 1 \\ 1 \cdot 10^{-5} & 1 \cdot 10^{-5} \end{pmatrix} \quad b = \begin{pmatrix} 1 \\ 2 \cdot 10^{-5} \end{pmatrix}$$

$$D_1 = \begin{pmatrix} 1 & 0 \\ 0 & 10^5 \end{pmatrix} \quad \text{calculada usando} \quad d_{1_i} = \left(\max_{1 \leq j \leq n} |a_{ij}| \right)^{-1}$$

$$D_1 A = \begin{pmatrix} 2 \cdot 10^{-5} & 1 \\ 1 & 1 \end{pmatrix} \quad \text{corresponde con la matriz } A_2 \text{ anterior donde} \\ |\det(D_1 A)| \approx 1$$

Finalmente resolvemos usando
eliminación gaussiana con pivoteo

$$D_1 A x = D_1 b \quad \Rightarrow \quad x = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

Obs. Esta estrategia de escalamiento puede ser incorporada al algoritmo de Gauss con pivoteo, lo cual lo hace más robusto. En ese caso, se escala cada fila de la matriz de coeficientes usando los elementos de la diagonal de D_1 .

Sistemas de Ecuaciones Lineales

Factorización LU general (otra manera)

$A = LU$ donde

$$A = \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{ij} & \dots & a_{in} \\ \vdots & \ddots & \vdots \\ a_{n1} & \dots & a_{nn} \end{pmatrix} \quad L = \begin{pmatrix} l_{11} & & & & \\ l_{21} & l_{22} & & & \\ l_{31} & l_{32} & l_{33} & & \\ \vdots & \vdots & \vdots & \ddots & \\ l_{n1} & l_{n2} & l_{n3} & \dots & l_{nn} \end{pmatrix} \quad U = \begin{pmatrix} u_{11} & u_{12} & u_{13} & \dots & u_{1n} \\ & u_{22} & u_{23} & \dots & u_{2n} \\ & & u_{33} & \dots & u_{3n} \\ & & & \ddots & \vdots \\ & & & & u_{nn} \end{pmatrix}$$

$l_{is} = 0$ para $s > i$ $u_{sj} = 0$ para $s > j$

$$a_{ij} = \left(l_{i1} \quad l_{i2} \quad l_{i3} \quad \dots \quad l_{in} \right) \begin{pmatrix} u_{1j} \\ u_{2j} \\ u_{3j} \\ \vdots \\ u_{nj} \end{pmatrix} = \sum_{s=1}^n l_{is} u_{sj} = \sum_{s=1}^{\min(i,j)} l_{is} u_{sj} \quad (14)$$

ceros $s > i$
ceros $s > j$

Sistemas de Ecuaciones Lineales

Factorización LU general (otra manera) (cont.)

$$a_{ij} = \sum_{s=1}^n l_{is} u_{sj} = \sum_{s=1}^{\min(i,j)} l_{is} u_{sj} \quad (14)$$

Fijamos $i=j=k$ en la ecuación (14) (elemento en la diagonal)

$$a_{kk} = \sum_{s=1}^{k-1} l_{ks} u_{sk} + l_{kk} u_{kk} \quad (15)$$

Supongamos que se han calculado los elementos de U hasta la fila $k-1$ y los elementos de L hasta la columna $k-1$, de (15) se tiene la relación

$$U = \begin{pmatrix} u_{11} & u_{12} & u_{13} & \cdots & u_{1n} \\ & u_{22} & u_{23} & \cdots & u_{2n} \\ & & u_{33} & \cdots & u_{3n} \\ & & & \ddots & \vdots \\ & & & & u_{nn} \end{pmatrix} \quad L = \begin{pmatrix} l_{11} & & & & \\ l_{21} & l_{22} & & & \\ l_{31} & l_{32} & l_{33} & & \\ \vdots & \vdots & \vdots & \ddots & \\ l_{n1} & l_{n2} & l_{n3} & \cdots & l_{nn} \end{pmatrix} \quad l_{kk} u_{kk} = a_{kk} - \sum_{s=1}^{k-1} l_{ks} u_{sk} \quad (16)$$

caso $k=3$

La relación (16) permite calcular u_{kk} o l_{kk} a partir del otro.

Sistemas de Ecuaciones Lineales

Factorización LU general (otra manera) (cont.)

A continuación con l_{kk} y u_{kk} calculados, procedemos a ubicar

- la fila k de U ($i=k$)
- la columna k de L ($j=k$)

Usando la ecuación (14) se tiene

$$\left. \begin{aligned} a_{kj} &= \sum_{s=1}^{k-1} l_{ks} u_{sj} + l_{kk} u_{kj} && \text{para } k+1 \leq j \leq n \\ a_{ik} &= \sum_{s=1}^{k-1} l_{is} u_{sk} + l_{ik} u_{kk} && \text{para } k+1 \leq i \leq n \end{aligned} \right\} \quad (17)$$

es decir, si $l_{kk} \neq 0$ y $u_{kk} \neq 0$

$$\left. \begin{aligned} u_{kj} &= \left(a_{kj} - \sum_{s=1}^{k-1} l_{ks} u_{sj} \right) / l_{kk} && \text{para } k+1 \leq j \leq n \\ l_{ik} &= \left(a_{ik} - \sum_{s=1}^{k-1} l_{is} u_{sk} \right) / u_{kk} && \text{para } k+1 \leq i \leq n \end{aligned} \right\} \quad (18)$$

$$U = \begin{pmatrix} u_{11} & u_{12} & u_{13} & \cdots & u_{1n} \\ & u_{22} & u_{23} & \cdots & u_{2n} \\ & & u_{33} & \cdots & u_{3n} \\ & & & \ddots & \vdots \\ & & & & u_{nn} \end{pmatrix}$$

caso $k=3$

$$L = \begin{pmatrix} l_{11} & & & & \\ l_{21} & l_{22} & & & \\ l_{31} & l_{32} & l_{33} & & \\ \vdots & \vdots & \vdots & \ddots & \\ l_{n1} & l_{n2} & l_{n3} & \cdots & l_{nn} \end{pmatrix}$$

Sistemas de Ecuaciones Lineales

Factorización LU general (otra manera) (cont.)

Obs. Es importante notar que los cálculos en (18) pueden realizarse en paralelo, lo cual puede representar un gran ahorro en tiempo de CPU.

Obs. El algoritmo basado en las fórmulas precedentes (16) y (18)

$$\left. \begin{aligned} l_{kk} u_{kk} &= a_{kk} - \sum_{s=1}^{k-1} l_{ks} u_{sk} \\ u_{kj} &= \left(a_{kj} - \sum_{s=1}^{k-1} l_{ks} u_{sj} + l_{kk} \right) / l_{kk} \quad \text{para } k+1 \leq j \leq n \\ l_{ik} &= \left(a_{ik} - \sum_{s=1}^{k-1} l_{is} u_{sk} + l_{kk} \right) / u_{kk} \quad \text{para } k+1 \leq i \leq n \end{aligned} \right\} 1 \leq k \leq n$$

se conoce como

- factorización LU de Doolittle cuando L es una matriz triangular inferior con 1 en la diagonal principal
- factorización LU de Crout cuando U es una matriz triangular superior con 1 en la diagonal principal
- factorización LU de Cholesky cuando $U = L^t$ (de donde $u_{kk} = l_{kk}$)

Sistemas de Ecuaciones Lineales

Factorización LU general (otra manera) (cont.)

Algoritmo {

Leer $A=(a_{ij}), n$

Para $k = 1$ hasta n

Especificar un valor no cero para l_{kk} o u_{kk} y

calcular el otro de

$$l_{kk} u_{kk} = a_{kk} - \sum_{s=1}^{k-1} l_{ks} u_{sk}$$

Para $j = k+1$ hasta n

$$u_{kj} = \left(a_{kj} - \sum_{s=1}^{k-1} l_{ks} u_{sj} \right) / l_{kk}$$

Fin para

Para $i = k+1$ hasta n

$$l_{ik} = \left(a_{ik} - \sum_{s=1}^{k-1} l_{is} u_{sk} \right) / u_{kk}$$

Fin para

Fin para

$$U = \begin{pmatrix} u_{11} & u_{12} & u_{13} & \cdots & u_{1n} \\ & u_{22} & u_{23} & \cdots & u_{2n} \\ & & u_{33} & \cdots & u_{3n} \\ & & & \ddots & \vdots \\ & & & & u_{nn} \end{pmatrix}$$

caso $k=3$

$$L = \begin{pmatrix} l_{11} & & & & \\ l_{21} & l_{22} & & & \\ l_{31} & l_{32} & l_{33} & & \\ \vdots & \vdots & \vdots & \ddots & \\ l_{n1} & l_{n2} & l_{n3} & \cdots & l_{nn} \end{pmatrix}$$

Sistemas de Ecuaciones Lineales

Factorización LU general (otra manera) (cont.)

Para resolver el sistema lineal $Ax = b$ usando la descomposición LU general de la matriz A , debemos resolver $LUX = b$, el cual lo operamos en 2 pasos (realizando el cambio $y = Ux$)

$$* Ly = b$$

$$* Ux = y$$

Algoritmo para resolver
 $LUX = b$

$$\left\{ \begin{array}{l} y_1 = b_1 / l_{11} \\ \text{Para } i = 2 \text{ hasta } n \\ y_i = \left(b_i - \sum_{j=1}^{i-1} l_{ij} y_j \right) / l_{ii} \\ \text{Fin para} \\ x_n = y_n / u_{nn} \\ \text{Para } i = n-1 \text{ hasta } 1 \\ x_i = \left(y_i - \sum_{j=i+1}^n u_{ij} x_j \right) / u_{ii} \\ \text{Fin para} \end{array} \right. \left. \begin{array}{l} \text{Resolver} \\ Ly=b \\ \\ \\ \text{Resolver} \\ Ux=y \end{array} \right.$$

Ejercicio: Calcular el número de operaciones básicas para resolver un sistema lineal de ecuaciones usando el método de descomposición LU general.

Sistemas de Ecuaciones Lineales

Método de descomposición Cholesky

Obs. Cuando una matriz A es simétrica, sería razonable esperar que en la factorización LU (cuando esta existe) de A , se tenga $U = L^T$.

En general esto no es cierto.

Obs. Supongamos A es no singular, $A = LL^T$ y $x \neq 0$, entonces L es no singular, y si definimos $y = L^T x \neq 0$, entonces

$$x^T Ax = x^T LL^T x = (L^T x)^T (L^T x) = y^T y = \sum_{i=1}^n y_i^2 > 0.$$

Entonces, una matriz A no singular que puede ser factorizada como LL^T tiene que ser definida positiva.

Sistemas de Ecuaciones Lineales

Método de descomposición Cholesky (cont.)

Obs. Sea A una matriz definida positiva entonces A es no singular.

Para ver esto, es suficiente demostrar que

$$Ax = 0 \Rightarrow x = 0$$

Supongamos lo contrario, es decir $Ax = 0$ pero $x \neq 0$.

Entonces, multiplicando por x^T por la izquierda

$$x^T Ax = 0$$


lo cual contradice el hecho que A es definida positiva.

Sistemas de Ecuaciones Lineales

Método de descomposición Cholesky (cont.)

Obs. Sea A una matriz $n \times n$ definida positiva de la forma $A = \begin{pmatrix} \alpha & a^T \\ a & A_* \end{pmatrix}$ donde a es un vector de longitud $n-1$, entonces $\alpha > 0$ y A_* es definida positiva.

Veamos que $\alpha > 0$. Sea $x^T = (1, 0, \dots, 0)$, entonces

$$0 < x^T A x = \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} \alpha & a^T \\ a & A_* \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \alpha \Rightarrow \alpha > 0.$$


vector nulo de longitud $n-1$

Veamos que A_* es definida positiva.

Sea $y \neq 0$ vector de dimensión $n-1$ y $x^T = (0 \ y^T)$, entonces

$$0 < x^T A x = \begin{pmatrix} 0 & y^T \end{pmatrix} \begin{pmatrix} \alpha & a^T \\ a & A_* \end{pmatrix} \begin{pmatrix} 0 \\ y \end{pmatrix} = y^T A_* y \Rightarrow y^T A_* y > 0.$$

Sistemas de Ecuaciones Lineales

Método de descomposición Cholesky (cont.)

Hemos visto que una matriz A no singular, la cual puede ser factorizada como LL^T , cumple que A es definida positiva.

El inverso es también verdad. Esto se enuncia a continuación.

Teorema:

Si A es una matriz simétrica y definida positiva, entonces

$$A = L L^T,$$

donde L es una matriz triangular inferior con los elementos de la diagonal positivos. Además esta descomposición es única.

Obs. La descomposición $A = L L^T$ recibe el nombre de **descomposición de Cholesky**.

Obs. Si $A = L L^T$ se tiene que $A = (-L) (-L^T)$ lo cual indica que A tiene otra descomposición de Cholesky. Esto no contradice el teorema anterior. ¿Por qué?



Sistemas de Ecuaciones Lineales

Método de descomposición Cholesky (cont.)

Para obtener el algoritmo de la descomposición de Cholesky, basta observar, en el algoritmo de la descomposición LU general, que:

lamina121

$$U = L^T$$

de donde, para cualquier índice k entre 1 y n , se cumple que:

$$L = \begin{pmatrix} l_{11} & & & & \\ l_{21} & l_{22} & & & \\ l_{31} & l_{32} & l_{33} & & \\ \vdots & \vdots & \vdots & \ddots & \\ l_{n1} & l_{n2} & l_{n3} & \cdots & l_{nn} \end{pmatrix}$$

$$u_{kk} = l_{kk}$$

$$u_{kj} = l_{jk}, \text{ para } k+1 \leq j \leq n$$

$$u_{ik} = l_{ki}, \text{ para } k+1 \leq i \leq n$$

$$U = \begin{pmatrix} u_{11} & u_{12} & u_{13} & \cdots & u_{1n} \\ & u_{22} & u_{23} & \cdots & u_{2n} \\ & & u_{33} & \cdots & u_{3n} \\ & & & \ddots & \vdots \\ & & & & u_{nn} \end{pmatrix}$$

Sistemas de Ecuaciones Lineales

Método de descomposición Cholesky (cont.)

Algoritmo

Leer $A=(a_{ij}), n$
 Para $k = 1$ hasta n

$$l_{kk} = \sqrt{a_{kk} - \sum_{s=1}^{k-1} l_{ks}^2}$$
 Para $i = k+1$ hasta n

$$l_{ik} = \left(a_{ik} - \sum_{s=1}^{k-1} l_{is} l_{ks} \right) / l_{kk}$$
 Fin para
 Fin para

Ejercicio: Calcular el número de operaciones básicas para resolver un sistema lineal de ecuaciones usando el método de descomposición de Cholesky.

Sistemas de Ecuaciones Lineales

Cálculo de la matriz inversa

Sea A es una matriz invertible, $A = (a_{ij})_{1 \leq i, j \leq n}$

Sean e_1, e_2, \dots, e_n la base canónica de R^n . Para el cálculo de la matriz inversa procedemos así

Calcular B tal que $AB = I$ con $B = \begin{pmatrix} b_{11} & \cdots & b_{1n} \\ \vdots & \ddots & \vdots \\ b_{n1} & \cdots & b_{nn} \end{pmatrix}$

Planteamos la secuencia de problemas siguientes

$$Ax_i = e_i \quad \text{con} \quad x_i = (b_{i1} \quad \cdots \quad b_{in})^T \quad \text{para} \quad 1 \leq i \leq n$$

(x_i es la columna i de la matriz B) por alguno de los procedimientos vistos.

Ejercicio: Calcular el número de operaciones básicas involucradas en la construcción de la inversa de A usando el método de descomposición LU .

Sistemas de Ecuaciones Lineales

Algoritmos para factorizar matrices especiales

Una matriz $A = (a_{ij})_{1 \leq i, j \leq n}$ se denomina matriz banda si existen enteros p y q , con la propiedad siguiente

$$a_{ij} = \begin{cases} 0 & \text{si } i + p \leq j \\ 0 & \text{si } j + q \leq i \end{cases} \quad \text{para } 1 < p, q < n$$

El ancho de banda es $w = p + q - 1$

Caso $p = q < n$

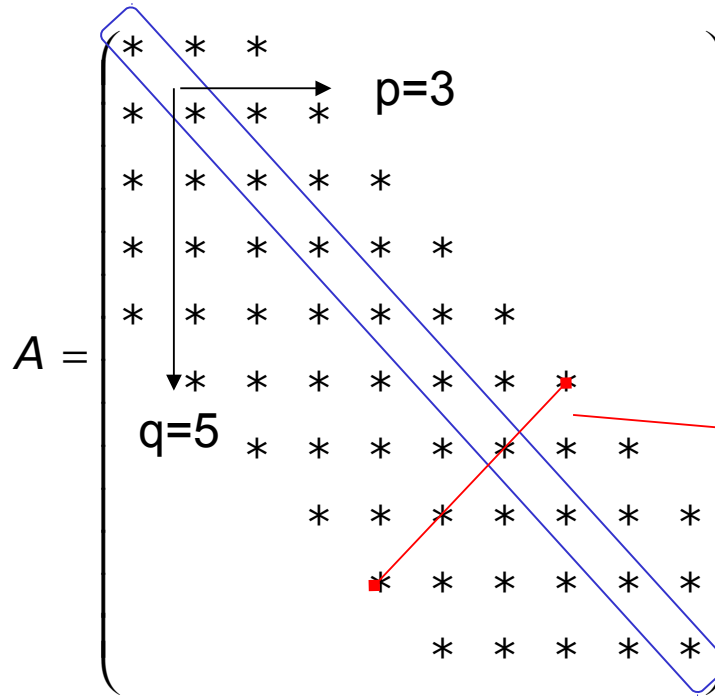
$$A = \begin{pmatrix} a_{11} & \cdots & a_{1p} & 0 & \cdots & 0 \\ \vdots & \ddots & & & \ddots & \vdots \\ a_{p1} & & a_{pp} & & & 0 \\ 0 & & & \ddots & & a_{pn} \\ \vdots & \ddots & & & & \vdots \\ 0 & \cdots & 0 & a_{np} & \cdots & a_{nn} \end{pmatrix}$$

$i+p \leq j$
 $j+p \leq i$

Un caso especial de estas matrices es cuando $p=q=2$, la cual da un ancho de banda $w=3$, y se denominan tridiagonales

Sistemas de Ecuaciones Lineales

Algoritmos para factorizar matrices especiales (cont.)



A es una matriz
banda con ancho
de banda

$$w = 3+5-1 = 7$$

$$a_{ij} = \begin{cases} 0 & \text{si } i + p \leq j \\ 0 & \text{si } j + q \leq i \end{cases} \quad \text{para } 1 < p, q < n$$

Sistemas de Ecuaciones Lineales

Algoritmos para factorizar matrices especiales (cont.)

Los algoritmos de factorización se pueden simplificar considerablemente en el caso de matrices banda, debido al gran número de ceros que aparecen en patrones regulares en estas matrices.

Caso A matriz tridiagonal

Supongamos que es posible encontrar matrices L y U , tal que $A = LU$.

$$L = \begin{pmatrix} l_{11} & 0 & \cdots & & 0 \\ l_{21} & l_{22} & \ddots & & \vdots \\ 0 & & \ddots & & \\ \vdots & \ddots & & & 0 \\ 0 & \cdots & 0 & l_{n,n-1} & l_{nn} \end{pmatrix} \quad U = \begin{pmatrix} 1 & u_{12} & 0 & \cdots & 0 \\ 0 & 1 & \ddots & & \vdots \\ & \ddots & \ddots & & 0 \\ \vdots & & & & u_{n-1,n} \\ 0 & \cdots & 0 & 0 & 1 \end{pmatrix}$$

La multiplicación $A = LU$ da, sin contar los elementos cero, las ecuaciones

$$\begin{array}{l|l} a_{11} = l_{11} & a_{ii} = l_{i,j-1}u_{i-1,j} + l_{ij} \quad \text{para } i = 2, \dots, n \\ a_{i,j-1} = l_{i,j-1} \quad \text{para } i = 2, \dots, n & a_{i,j+1} = l_{ij}u_{i,j+1} \quad \text{para } i = 1, \dots, n-1 \end{array}$$

Sistemas de Ecuaciones Lineales

Algoritmos para factorizar matrices especiales (cont.)

Algoritmo para matrices tridiagonales	$\text{Leer } A = (a_{ij}), n$ $l_{11} = a_{11}; \quad u_{12} = a_{12} / l_{11}$ $\text{Para } i = 2 \text{ hasta } n-1$ $l_{i,j-1} = a_{i,j-1}$ $l_{ii} = a_{ii} - l_{i,j-1} u_{i-1,i}$ $u_{i,j+1} = a_{i,j+1} / l_{ii}$	$\left. \begin{array}{l} \\ \\ \\ \end{array} \right\} \begin{array}{l} \text{i-ésima fila de } L \\ \\ \text{(i+1) columna de } U \end{array}$
	Fin para $l_{n,n-1} = a_{n,n-1}$ $l_{nn} = a_{nn} - l_{n,n-1} u_{n-1,n}$	$\left. \begin{array}{l} \\ \\ \end{array} \right\} \text{n-ésima fila de } L$

Ejercicio:

- Calcular el número de operaciones elementales involucradas en el algoritmo.
- Modificar el algoritmo anterior usando un conjunto de vectores para las matrices A , L y U (no almacenar los ceros).

Sistemas de Ecuaciones Lineales

Algoritmos para factorizar matrices especiales (cont.)

Una matriz $A = (a_{ij})_{1 \leq i, j \leq n}$ se denomina de Hessenberg superior si para todo $i > j + 1$ se tiene $a_{ij} = 0$

Para el caso que
la dimensión de
A sea 5×5

$$A = \begin{pmatrix} * & * & * & * & * \\ * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & 0 & * & * \end{pmatrix}$$

Apliquemos Eliminación Gaussiana a esta matriz.

En el primer paso, sólo el elemento (2,1) requiere ser eliminado, el resto en la primera columna ya son ceros.

Para esto tenemos que restar un múltiplo de la primera fila de la segunda para obtener la matriz

$$\begin{pmatrix} * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & 0 & * & * \end{pmatrix}$$

Sistemas de Ecuaciones Lineales

Algoritmos para factorizar matrices especiales (cont.)

En el segundo paso, restamos un múltiplo de la segunda fila de la tercera

$$\begin{pmatrix} * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & 0 & * & * \end{pmatrix}$$

Dos pasos más del proceso llevan a las matrices

$$\begin{pmatrix} * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & 0 & * & * \\ 0 & 0 & 0 & * & * \end{pmatrix} \quad y \quad \begin{pmatrix} * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & 0 & * & * \\ 0 & 0 & 0 & 0 & * \end{pmatrix}$$

siendo la última triangular superior.

Sistemas de Ecuaciones Lineales

Algoritmos para factorizar matrices especiales (cont.)

El pseudo-código para reducir este tipo de matrices a triangular superior, dejando de lado el pivoteo por simplicidad, es el siguiente:

Algoritmo de
triangularización
para matrices
Hessenberg
superior

Leer $A = (a_{ij})$, n

Para $k = 1$ hasta $n-1$

$$a(k+1,k) = a(k+1,k)/a(k,k)$$

Para $j = k+1$ hasta n

$$a(k+1,j) = a(k+1,j) - a(k+1,k) * a(k,j)$$

Fin para

Fin para

Comparar con
el método de
eliminación
gaussiana, p.90

hess2triangular.m para el
caso de almacenamiento
en 2 vectores.

Los factores de multiplicación sobrescriben los elementos debajo de la diagonal principal de A , y la matriz triangular final sobrescribe el triángulo superior de A .

Ejercicio: Calcular el número de operaciones elementales involucradas en este algoritmo y comparar con Gauss. Incluir pivoteo.

Sistemas de Ecuaciones Lineales

Técnicas de mejoramiento de la solución. Refinamiento Iterativo.

Sea $proc(A,b)$ un procedimiento que calcula el vector \bar{x} a partir de una matriz A y un vector b , es decir el vector \bar{x} es solución para $Ax=b$,

$$\bar{x} \leftarrow proc(A,b).$$

Luego si x es la solución exacta de $Ax=b$, entonces podemos definir Δb

$$\Delta b = A\bar{x} - b = A\bar{x} - Ax = A\Delta x \quad \text{con} \quad \Delta x = \bar{x} - x.$$

Podemos aplicar el procedimiento anterior para resolver $A\Delta x = \Delta b$

$$\overline{\Delta x} \leftarrow proc(A, \Delta b),$$

así, $\overline{\Delta x}$ es una aproximación del verdadero error Δx .

Podemos decir que la nueva solución será $\bar{\bar{x}} = \bar{x} + \overline{\Delta x}$, que es una mejora de la solución \bar{x} .

Este procedimiento lo podemos repetir varias veces!

Sistemas de Ecuaciones Lineales

Técnicas de mejoramiento de la solución. Refinamiento Iterativo (cont.)

Cuántas veces debemos repetir este procedimiento?

Para detenerlo procedemos así, dado ε muy pequeño

a) ver si $\|A\bar{x} - b\| < \varepsilon$

b) ver si $\|\Delta x\| < \varepsilon$

c) o una combinación de (a) y (b) al mismo tiempo

Ejercicio: Escribir un programa que lleve a cabo el mejoramiento iterativo de la solución aproximada x del sistema lineal de ecuaciones $Ax = b$.

Sistemas de Ecuaciones Lineales

De respuestas a las siguientes interrogantes:

- (1) ¿Toda matriz tiene al menos una descomposición LU ?
- (2) ¿Toda matriz invertible tiene al menos una descomposición LU ?
- (3) ¿Una matriz que tiene uno o más ceros en su diagonal principal puede tener descomposición LU ?
- (4) Si una matriz A tiene descomposición de Cholesky, $A=LL^T$, donde L es una matriz triangular inferior, ¿se puede asegurar que A es definida positiva?
- (5) ¿Toda matriz diagonal dominante estricta es definida positiva?
- (6) ¿Toda matriz definida positiva es diagonal dominante estricta?



Sistemas de Ecuaciones Lineales

Respuestas a las interrogantes planteadas:

(1) (Falso). Si la matriz $A = \begin{pmatrix} 0 & 1 \\ 1 & 2 \end{pmatrix}$

tuviera una descomposición LU , se tendría algo así:

$$\begin{pmatrix} 0 & 1 \\ 1 & 2 \end{pmatrix} = \begin{pmatrix} a & 0 \\ b & c \end{pmatrix} \begin{pmatrix} d & e \\ 0 & f \end{pmatrix}$$

$$A = L \cdot U$$

de donde

$$\begin{cases} ad = 0 \\ ae = 1 \\ bd = 1 \\ be + cf = 2 \end{cases}$$

Así, $a=0$ ó $d=0$. Cualquiera de estas dos posibilidades genera una contradicción con las ecuaciones $ae=1$ y $bd=1$. Por lo tanto, la matriz en cuestión no puede tener una descomposición LU .

Sistemas de Ecuaciones Lineales

Respuestas a las interrogantes planteadas:

(2) (Falso). La matriz $A = \begin{pmatrix} 0 & 1 \\ 1 & 2 \end{pmatrix}$ es invertible,

ya que su determinante es $-1 \neq 0$.

Sin embargo, ya se probó en la lámina anterior que dicha matriz no tiene ninguna descomposición LU .

Sistemas de Ecuaciones Lineales

Descomposición QR de una matriz

Sea L una recta en R^2 que pasa por el origen. El operador Q que refleja todo vector de R^2 a través de la recta L es una transformación lineal, la cual puede ser representada por una matriz.

u y $v \in R^2$ con $\|u\| = 1$,

v el vector dirección de la recta L

y u y v perpendiculares.

u es perpendicular a $v \Leftrightarrow u^t v = 0$

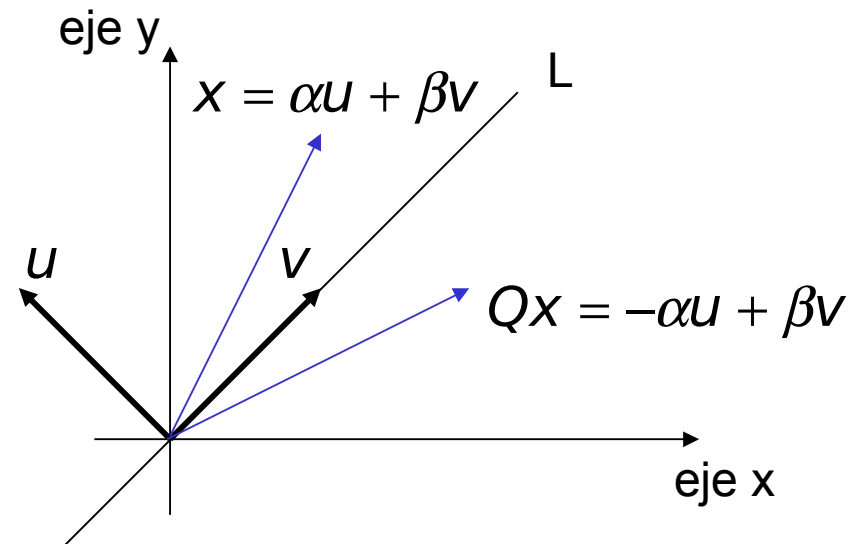
u y v forman un base de R^2

así, si $x \in R^2 \Rightarrow x = \alpha u + \beta v$

La reflexión de X a través de la recta L es $-\alpha u + \beta v$

La reflexión de u a través de la recta L es $-u$

La reflexión de v a través de la recta L es v



Sistemas de Ecuaciones Lineales

Descomposición QR de una matriz (cont.)

Podemos construir la matriz $P = \vec{u} \vec{u}^t \in R^{2 \times 2}$

Se verifica:

$$P\vec{u} = (\vec{u} \vec{u}^t) \vec{u} = \vec{u} (\vec{u}^t \vec{u}) = \vec{u} \|\vec{u}\|^2 = \vec{u}$$

$$P\vec{v} = (\vec{u} \vec{u}^t) \vec{v} = \vec{u} (\vec{u}^t \vec{v}) = \vec{u} 0 = 0$$

Construimos la matriz $Q = I - 2P = I - 2\vec{u} \vec{u}^t \in R^{2 \times 2}$

Se verifica:

$$Q\vec{u} = \vec{u} - 2P\vec{u} = \vec{u} - 2\vec{u} = -\vec{u}$$

$$Q\vec{v} = \vec{v} - 2P\vec{v} = \vec{v}$$

Para $x = \alpha\vec{u} + \beta\vec{v} \in R^2$ se tiene

$$Qx = Q(\alpha\vec{u} + \beta\vec{v}) = \alpha Q(\vec{u}) + \beta Q(\vec{v}) = -\alpha\vec{u} + \beta\vec{v}$$

Q es la matriz que refleja vectores a través de la recta L.

Sistemas de Ecuaciones Lineales

Descomposición QR de una matriz (cont.)

En resumen para

$$u \in R^n, \|u\| = 1, P = u u^t \in R^{n \times n}, Q = I - 2P \in R^{n \times n}$$

se tiene

$$1) \quad Pu = u, \quad Pv = 0 \quad \text{si } u^t v = 0, \quad P^2 = P, \quad P^t = P \quad (\text{simétrica})$$

$$2) \quad Qu = -u, \quad Qv = v \quad \text{si } u^t v = 0, \quad Q^2 = I,$$

$$Q = Q^t \quad (\text{simétrica}), \quad Q^{-1} = Q^t \quad (\text{ortogonal})$$

Definición. Si $u \in R^n, \|u\| = 1, Q = I - 2u u^t$
entonces Q se denomina un **reflector o transformación de Householder**.

Sistemas de Ecuaciones Lineales

Descomposición QR de una matriz (cont.)

Teorema 1. Si $u \in R^n$, $u \neq 0$, $\gamma = 2/\|u\|_2^2$, $Q = I - \gamma uu^t$ entonces Q es un reflector o transformación de Householder.

Teorema 2. Si x e $y \in R^n$, $x \neq y$, $\|x\|_2 = \|y\|_2$, entonces existe un reflector o transformación de Householder Q , tal que $Qx = y$.

Teorema 3. Reflectores pueden ser usados para crear ceros en vectores y matrices, es decir, para

$$x \in R^n, x \neq 0, \sigma = \pm\|x\|, y = (-\sigma, 0, \dots, 0)^t,$$

existe un reflector Q tal que $Qx = y$.

Teorema 4. Sea $A \in R^{n \times n}$, esta puede ser expresada como $A = QR$ donde Q es ortogonal y R triangular superior.

Sistemas de Ecuaciones Lineales

Descomposición QR de una matriz (cont.)

Ejemplo. Encontrar la descomposición QR para $A = \begin{pmatrix} 1 & 2 \\ 1 & 3 \end{pmatrix}$

$$Q = \begin{pmatrix} -0.7071 & -0.7071 \\ -0.7071 & 0.7071 \end{pmatrix} \quad R = \begin{pmatrix} -1.4142 & -3.5355 \\ 0 & 0.7071 \end{pmatrix}$$

$Q * Q^T = Q^T * Q = I$ R es triangular superior

En Matlab:
[Q,R] = qr(A)

Ejemplo. Si $B = \text{hilb}(5)$, determinar $\det(B)$, $\text{cond}(B)$ y las descomposiciones LU y QR de B, si esto es posible.

Obs.

- Toda matriz $n \times n$ tiene descomposición QR. Lo mismo no puede decirse para la existencia de su descomposición LU.
- Dada la descomposición $A = QR$, resolver $Ax = b$ se reduce a resolver $Rx = Qb$ (probarlo). Esto se lleva a cabo usando sustitución hacia atrás.
- El costo de aplicar la descomposición QR es “alto” (este requiere el doble del número de operaciones elementales del método de descomposición LU).



Sistemas de Ecuaciones Lineales

Métodos iterativos (métodos estacionarios)

Dado un sistema lineal $Ax = b$ (1)

Def. Se denomina **método iterativo de resolución** a aquel que genera una sucesión de vectores $x^{(i)}$ ($i = 0, 1, 2, \dots$) a partir de un vector dado $x^{(0)}$.

Forma de los métodos considerados $x^{(i+1)} = H x^{(i)} + c$. (2)

Def. Se dice que un método iterativo de resolución (2) es **consistente** con el sistema (1) si $x = H x + c$, donde x es la solución de (1).

Def. Se dice que un método iterativo de resolución del sistema (1) es **convergente** si para todo vector inicial $x^{(0)}$ se verifica:

$$\lim_{i \rightarrow \infty} x^{(i)} = x = A^{-1}b.$$

Obs. Los conceptos de consistencia y de convergencia deben distinguirse. Todo método convergente es consistente, pero no todo método consistente debe ser necesariamente convergente.

Sistemas de Ecuaciones Lineales

Métodos iterativos

Ejemplo. Sea el método iterativo $x^{(i+1)} = 3x^{(i)} - 2A^{-1}b$.

Es evidente que el método es consistente con (1).

Para $x = A^{-1}b$ se tiene $x^{(i+1)} - x = 3x^{(i)} - 2A^{-1}b - x = 3(x^{(i)} - x)$,

en consecuencia, el límite de la sucesión $x^{(i)}$, si existe, debe ser x (ya que si fuese otro vector z diferente de x , se debería verificar que

$$z - x = 3(z - x)$$

lo cual es un absurdo).

No obstante es evidente que $\lim_{i \rightarrow \infty} x^{(i)} \neq x$, salvo para el caso $x^{(0)} = x$

(ya que para cualquier otro vector $x^{(0)}$ distinto de x , el vector diferencia entre $x^{(i+1)}$ y x es 3 veces el vector diferencia entre $x^{(i)}$ y x , lo que indica que dicho vector diferencia aumenta con i en vez de tender hacia 0 como ocurriría si convergiese la sucesión hacía el único límite posible que acabamos de indicar que es x).

Por lo tanto el método no es convergente.

Sistemas de Ecuaciones Lineales

Métodos iterativos

Los métodos iterativos que vamos a considerar consisten en descomponer la matriz A de la forma

$$A = M - N, \quad (2)$$

donde A , M , N son matrices $n \times n$.

Reemplazando (2) en (1) se obtiene

$$(M - N)x = b \quad \Rightarrow \quad Mx = Nx + b.$$

Ahora, supongamos que M es invertible, entonces

$$x = M^{-1}Nx + M^{-1}b.$$

Se propone el método iterativo siguiente

$$\boxed{\begin{aligned} x^{(i+1)} &= Hx^{(i)} + c \\ \text{con } H &= M^{-1}N \quad \text{y} \quad c = M^{-1}b. \end{aligned}} \quad (3)$$

Sistemas de Ecuaciones Lineales

Métodos iterativos

Obs. Si la sucesión $\{x^{(i)}\}$ definida en (3) converge a un vector y , entonces y es la solución de (1) (el método es consistente con (1)). Prueba.

Dado que (3) es cierto para todos los índices i , se puede tomar límite a ambos lados de dicha ecuación, lo cual nos dice que:

$$\lim_{i \rightarrow \infty} x^{(i+1)} = \lim_{i \rightarrow \infty} (Hx^{(i)} + c)$$

Como $\lim_{i \rightarrow \infty} x^{(i+1)} = \lim_{i \rightarrow \infty} x^{(i)}$ (es la misma sucesión) se tiene que:

$$\lim_{i \rightarrow \infty} x^{(i+1)} = H \left(\lim_{i \rightarrow \infty} x^{(i)} \right) + c \Rightarrow y = Hy + c$$

$$\Rightarrow y = M^{-1}Ny + M^{-1}b \Rightarrow My = Ny + b$$

$$\Rightarrow (M - N)y = b \Rightarrow Ay = b$$



Sistemas de Ecuaciones Lineales

Método de Richardson

Sea A es una matriz invertible, $A = (a_{ij})_{1 \leq i, j \leq n}$ resolver $Ax = b$.

Sea $M = I$, $N = M - A = I - A$

Así, la iteración (3) se escribe como

$$\begin{aligned} x^{(i+1)} &= (I - A)x^{(i)} + b = x^{(i)} + b - Ax^{(i)} \\ x^{(i+1)} &= x^{(i)} + r^{(i)} \end{aligned} \quad (4)$$

donde $r^{(i)} = b - Ax^{(i)}$ es el vector residual.

Obs. Para detener el proceso iterativo se pueden utilizar los siguientes criterios:

a) cuando el número de iteraciones alcanza un número máximo

b) dado ε , cuando $\|x^{(i+1)} - x^{(i)}\| < \varepsilon$ o $\|b - Ax^{(i)}\| < \varepsilon$

(ε es la tolerancia)

Sistemas de Ecuaciones Lineales

Método de Richardson (cont.)

Leer $A=(a_{ij}), b, n, x, \varepsilon, maxit$

Para $k = 1$ hasta $maxit$

Para $i = 1$ hasta n

$$r_i = b_i - \sum_{j=1}^n a_{ij} x_j$$

Fin para

$norma_r = abs(r_1)$

Para $i = 2$ hasta n

$s = abs(r_i)$

Si $s > norma_r$ entonces

$norma_r = s$

Fin si

Fin para

Si $norma_r < \varepsilon$

break

Fin si

Para $i = 1$ hasta n

$x_i = x_i + r_i$

Fin para

Fin para Prof. Saúl Buitrago y Oswaldo Jiménez

construcción del
vector residual

cálculo de la
norma inf del
vector residual

detener si la norma
del vector residual es
menor que ε

nueva
aproximación

Algoritmo
de
Richardson

Sistemas de Ecuaciones Lineales

Ejemplo: Iteración de Richardson $x^{(i+1)} = (I - A)x^{(i)} + b$

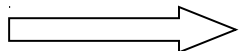
Resolver el sistema

$$\begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{3} \\ \frac{1}{3} & 1 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{3} & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} \frac{11}{18} \\ \frac{11}{18} \\ \frac{11}{18} \end{pmatrix}$$

$$A = [1, 1/2, 1/3; 1/3, 1, 1/2; 1/2, 1/3, 1]$$

$$b = [11/18; 11/18; 11/18]$$

usando $x^0 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$ $\|x^i - x^{i-1}\| \leq eps$



$$\left\{ \begin{array}{l} x^1 = (0.611 \quad 0.611 \quad 0.611)^T \\ \dots \\ x^{10} = (0.279 \quad 0.279 \quad 0.279)^T \\ \dots \\ x^{198} = (0.333 \quad 0.333 \quad 0.333)^T \end{array} \right.$$

Notar que $I - A = \begin{pmatrix} 0 & -\frac{1}{2} & -\frac{1}{3} \\ -\frac{1}{3} & 0 & -\frac{1}{2} \\ -\frac{1}{2} & -\frac{1}{3} & 0 \end{pmatrix} \Rightarrow \|I - A\|_{\infty} = \frac{5}{6} < 1$

Sistemas de Ecuaciones Lineales

Método de Jacobi

Sea A es una matriz invertible, $A = (a_{ij})_{1 \leq i, j \leq n}$ se quiere resolver $Ax = b$.

Sea $A = D - E - F$

$E = (e_{ij})_{1 \leq i, j \leq n}$ con

$$e_{ij} = \begin{cases} 0 & \text{si } i \leq j \\ -a_{ij} & \text{si } i > j \end{cases}$$

- triángulo inferior de A
(sin la diagonal principal)

$F = (f_{ij})_{1 \leq i, j \leq n}$ con

$$f_{ij} = \begin{cases} 0 & \text{si } i \geq j \\ -a_{ij} & \text{si } i < j \end{cases}$$

- triángulo superior de A
(sin la diagonal principal)

$D = (d_{ij})_{1 \leq i, j \leq n}$ con

$$d_{ij} = \begin{cases} 0 & \text{si } i \neq j \\ a_{ij} & \text{si } i = j \end{cases}$$

diagonal principal de A

Supongamos que los elementos de a_{ij} , $i=1, \dots, n$ son no nulos.

Definimos $M=D$ y $N=E+F$.

Así, se tiene el método iterativo

$$x^{(i+1)} = \underbrace{D^{-1}(E + F)}_H x^{(i)} + \underbrace{D^{-1}b}_c \quad (5)$$

Obs. Para detener el proceso iterativo se procede igual que en el método de Richardson.

Sistemas de Ecuaciones Lineales

Método de Jacobi (cont.)

Leer $A=(a_{ij}), b, n, xi, \varepsilon$

Para $i = 1$ hasta n

$$b_i = b_i / a_{ii}$$

Fin para

Para $i = 2$ hasta $n-1$

Para $j = 1$ hasta $i-1$

$$a_{ij} = -a_{ij} / a_{ii}$$

Fin para

Para $j = i+1$ hasta n

$$a_{ij} = -a_{ij} / a_{ii}$$

Fin para

Fin para

Para $j = 2$ hasta n

$$a_{1j} = -a_{1j} / a_{11}$$

$$a_{n,j-1} = -a_{n,j-1} / a_{nn}$$

Fin para

$vecdif = 5$

Mientras $vecdif > \varepsilon$

Para $i = 2$ hasta $n-1$

$$s = 0$$

Para $j = 1$ hasta $i-1$

$$s = s + a_{ij} * xi_j$$

Fin para

Para $j = i+1$ hasta n

$$s = s + a_{ij} * xi_j$$

Fin para

$$xi_i = s + b_i$$

Fin para

$$s = 0; ss = 0$$

Para $j = 2$ hasta n

$$s = s + a_{1j} * xi_j$$

$$ss = ss + a_{n,j-1} * xi_{j-1}$$

Fin para

$$xi_1 = s + b_1$$

$$xi_n = ss + b_n$$

$$vecdif = abs(xi_1 - xi_1)$$

Para $i = 2$ hasta n

$$s = abs(xi_i - xi_i)$$

Si $s > vecdif$

$$vecdif = s$$

Fin si

Fin para

Fin mientras

Obs.

- No se detiene el algoritmo por número máximo de iteraciones.
- No se cuenta el número de iteraciones realizadas

Sistemas de Ecuaciones Lineales

Método de Jacobi (cont.)

El algoritmo del método de Jacobi, mostrado en la lámina anterior, se puede simplificar de la siguiente manera:

Partiendo de la ecuación (5) $x^{(i+1)} = D^{-1}(E + F)x^{(i)} + D^{-1}b$ se tiene que:

$$Dx^{(i+1)} = (E + F)x^{(i)} + b$$

La fila j -ésima de este sistema es:

$$a_{jj}x_j^{(i+1)} = -\sum_{\substack{k=1 \\ k \neq j}}^n a_{jk}x_k^{(i)} + b_j \quad (j = 1, 2, \dots, n)$$

de donde

$$x_j^{(i+1)} = \frac{1}{a_{jj}} \left[b_j - \sum_{\substack{k=1 \\ k \neq j}}^n a_{jk}x_k^{(i)} \right], \quad (j = 1, 2, \dots, n).$$

siempre que $a_{jj} \neq 0$, para $j = 1, 2, \dots, n$.

Sistemas de Ecuaciones Lineales

Método de Jacobi (cont.)

Con esta última ecuación se puede escribir el algoritmo del método de Jacobi como se muestra a continuación:

```

Leer  $A=(a_{ij}), b, n, x, \varepsilon, maxit$ 
Para  $k = 1$  hasta  $maxit$ 
  Para  $j = 1$  hasta  $n$ 
    suma = 0
    Para  $k = 1$  hasta  $n$ 
      Si  $k \neq j$  entonces
        suma = suma +  $a(j,k) * x(k)$ 
      Fin si
    Fin para
     $z(j) = (b(j) - suma) / a(j,j)$ 
  Fin para
  Si  $abs(z - x) < \varepsilon$ 
    return
  Fin si
   $x = z$ 
Fin para

```

Sistemas de Ecuaciones Lineales

Ejemplo: Iteración de Jacobi $x^{(i+1)} = D^{-1}(E + F)x^{(i)} + D^{-1}b$

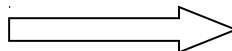
Resolver el sistema

$$\begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{3} \\ \frac{1}{3} & 1 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{3} & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} \frac{11}{18} \\ \frac{11}{18} \\ \frac{11}{18} \end{pmatrix}$$

$$A = [1, 1/2, 1/3; 1/3, 1, 1/2; 1/2, 1/3, 1]$$

$$b = [11/18; 11/18; 11/18]$$

usando $x^0 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$ $\|x^i - x^{i-1}\| \leq eps$



$$x^{198} = (0.333 \quad 0.333 \quad 0.333)^T$$

Obs. En este ejemplo la matriz de iteración coincide con la de Richardson.

Notar que $D^{-1}(E + F) = \begin{pmatrix} 0 & -\frac{1}{2} & -\frac{1}{3} \\ -\frac{1}{3} & 0 & -\frac{1}{2} \\ -\frac{1}{2} & -\frac{1}{3} & 0 \end{pmatrix} \Rightarrow \|D^{-1}(E + F)\|_{\infty} = \frac{5}{6} < 1$

Sistemas de Ecuaciones Lineales

Método de Gauss-Seidel

Sea A es una matriz invertible, $A = (a_{ij})_{1 \leq i, j \leq n}$ se quiere resolver $Ax = b$.

Sea $A = D - E - F$, con D, E, F como en el método de Jacobi, para el método iterativo(3) se toma

$$M = D - E \quad \text{y} \quad N = F$$

y se tiene el método de Gauss-Seidel

$$(D - E)x^{(i+1)} = Fx^{(i)} + b \Leftrightarrow x^{(i+1)} = \boxed{(D - E)^{-1} F} x^{(i)} + \boxed{(D - E)^{-1} b} \quad (6)$$

el cual, representado en forma matricial es

$$\underbrace{\begin{pmatrix} a_{11} & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ & & 0 & \\ a_{n1} & \cdots & & a_{nn} \end{pmatrix}}_{D-E} \begin{pmatrix} x_1^{(i+1)} \\ \vdots \\ x_n^{(i+1)} \end{pmatrix} = \underbrace{\begin{pmatrix} 0 & -a_{12} & \cdots & -a_{1n} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & & & -a_{n-1,n} \\ 0 & \cdots & & 0 \end{pmatrix}}_F \begin{pmatrix} x_1^{(i)} \\ \vdots \\ x_n^{(i)} \end{pmatrix} + \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}$$

Sistemas de Ecuaciones Lineales

Método de Gauss-Seidel

En (6) no hay que invertir la matriz $D-E$, se procede como en el método de sustitución hacia adelante, es decir

$$x_1^{(i+1)} = \left(-\sum_{j=2}^n a_{1j} x_j^{(i)} + b_1 \right) / a_{11}$$

$$x_2^{(i+1)} = \left(-a_{21} x_1^{(i+1)} - \sum_{j=3}^n a_{2j} x_j^{(i)} + b_2 \right) / a_{22}$$

...

$$x_{n-1}^{(i+1)} = \left(-\sum_{j=1}^{n-2} a_{n-1,j} x_j^{(i+1)} - a_{n-1,n} x_n^{(i)} + b_{n-1} \right) / a_{n-1,n-1}$$

$$x_n^{(i+1)} = \left(-\sum_{j=1}^{n-1} a_{nj} x_j^{(i+1)} + b_n \right) / a_{nn}$$

Obs. Notemos que esto corresponde a aprovechar en el paso k , los valores $x_j^{(j+1)}$, $j = 1, \dots, k-1$ ya calculados.

Ejercicio. Escribir el algoritmo de Gauss-Seidel utilizando el criterio de parada (b) ($\|x^{(i+1)} - x^{(i)}\| < \varepsilon$) y usando la iteración $(D - E)x^{(i+1)} = Fx^{(i)} + b$.

Sistemas de Ecuaciones Lineales

Ejemplo: Iteración de Gauss-Seidel $x^{(i+1)} = (D - E)^{-1} Fx^{(i)} + (D - E)^{-1} b$

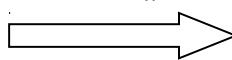
Resolver el sistema

$$\begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{3} \\ \frac{1}{3} & 1 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{3} & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} \frac{11}{18} \\ \frac{11}{18} \\ \frac{11}{18} \end{pmatrix}$$

$$A = [1, 1/2, 1/3; 1/3, 1, 1/2; 1/2, 1/3, 1]$$

$$b = [11/18; 11/18; 11/18]$$

usando $x^0 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$ $\|x^i - x^{i-1}\| \leq eps$



$$x^{36} = (0.333 \quad 0.333 \quad 0.333)^T$$

Notar que $(D - E)^{-1} F = \begin{pmatrix} 0 & -0.5 & -0.333 \\ 0 & 0.167 & -0.389 \\ 0 & -0.194 & 0.296 \end{pmatrix} \Rightarrow \|(D - E)^{-1} F\|_{\infty} = 0.833 < 1$

Sistemas de Ecuaciones Lineales

Métodos de Relajación Sucesiva (SOR: successive over relaxation)

Se quiere resolver el sistema $AX = b$, con A invertible, $A = D - E - F$, siendo D la diagonal de A , $-E$ el triángulo inferior y $-F$ el triángulo superior.

Si $\omega > 0$, se tiene

$$\omega AX = \omega b \Leftrightarrow \{-\omega E + \omega D - \omega F\}X = \omega b$$

sumando y restando D

$$\Leftrightarrow \{D - \omega E - (1 - \omega)D - \omega F\}X = \omega b$$

$$\Leftrightarrow \{D - \omega E\}X = \{\omega F + (1 - \omega)D\}X + \omega b.$$

Se propone la iteración

$$\underbrace{\{D - \omega E\}}_M X^{(i+1)} = \underbrace{\{\omega F + (1 - \omega)D\}}_N X^{(i)} + \omega b \quad (7)$$

$$MX^{(i+1)} = NX^{(i)} + \omega b \quad (7a)$$

Sistemas de Ecuaciones Lineales

Métodos de Relajación Sucesiva (cont.)

Debemos tener M invertible y no se impone ninguna condición sobre N .

$$M = D - \omega E = \begin{pmatrix} a_{11} & 0 & \cdots & 0 \\ \omega a_{21} & \ddots & \ddots & \vdots \\ \vdots & & & 0 \\ \omega a_{n1} & \cdots & & a_{nn} \end{pmatrix} \begin{matrix} \rightarrow D \\ \\ \\ \rightarrow -\omega E \end{matrix}$$

$$N = \omega F + (1 - \omega)D = \begin{pmatrix} (1 - \omega)a_{11} & -\omega a_{12} & \cdots & -\omega a_{1n} \\ 0 & \ddots & \ddots & \vdots \\ \vdots & & & \\ 0 & \cdots & & (1 - \omega)a_{nn} \end{pmatrix} \begin{matrix} \rightarrow \omega F \\ \\ \\ \rightarrow (1 - \omega)D \end{matrix}$$

$$Mx^{(i+1)} = Nx^{(i)} + \omega b$$

El nuevo vector iterado se calcula como en el método de Gauss-Seidel, usando (7A), donde no hace falta calcular la inversa de $M = D - \omega E$.

Sistemas de Ecuaciones Lineales

Métodos de Relajación Sucesiva (cont.)

Obs. Según la escogencia de ω el método recibe los nombres de:

- Si $0 < \omega < 1$ se denomina de sub-relajación
- Si $\omega > 1$ se denomina de sobre-relajación

Obs.

- La iteración (7) se conoce como SOR hacia delante: se tomó

$$M = D - \omega E \quad y \quad N = \omega F + (1-\omega)D.$$

- Si tomamos en (7a)

$$M = D - \omega F \quad y \quad N = \omega E + (1-\omega)D$$

el método se denomina SOR hacia atrás.

Ejercicio. Escribir el algoritmo SOR hacia delante utilizando el criterio de parada (b) ($\|x^{(i+1)} - x^{(i)}\| < \varepsilon$).

Sistemas de Ecuaciones Lineales

Comparación de métodos iterativos

$$Ax = b$$

$$A = D - E - F = M - N,$$

$$x^{(i+1)} = Hx^{(i)} + c$$

$$\text{con } H = M^{-1}N \quad \text{y} \quad c = M^{-1}b.$$

$$Mx^{(i+1)} = Nx^{(i)} + b$$

Nombre del método	Jacobi	Gauss-Seidel	SOR
Descomposición $A=M-N$	$A = D - (E + F)$	$A = (D - E) - F$	$A = \left(\frac{D}{\omega} - E\right) - \left(\frac{1-\omega}{\omega}D + F\right)$
Matriz $H=M^{-1}N=I-M^{-1}A$ método iterativo	$D^{-1}(E + F) = I - D^{-1}A$	$(D - E)^{-1}F$	$\left(\frac{D}{\omega} - E\right)^{-1} \left(\frac{1-\omega}{\omega}D + F\right)$
Descripción de una iteración	$Dx^{(k+1)} = (E + F)x^{(k)} + b$	$(D - E)x^{(k+1)} = Fx^{(k)} + b$	$(D - \omega E)x^{(k+1)} = ((1 - \omega)D + \omega F)x^{(k)} + \omega b$

Sistemas de Ecuaciones Lineales

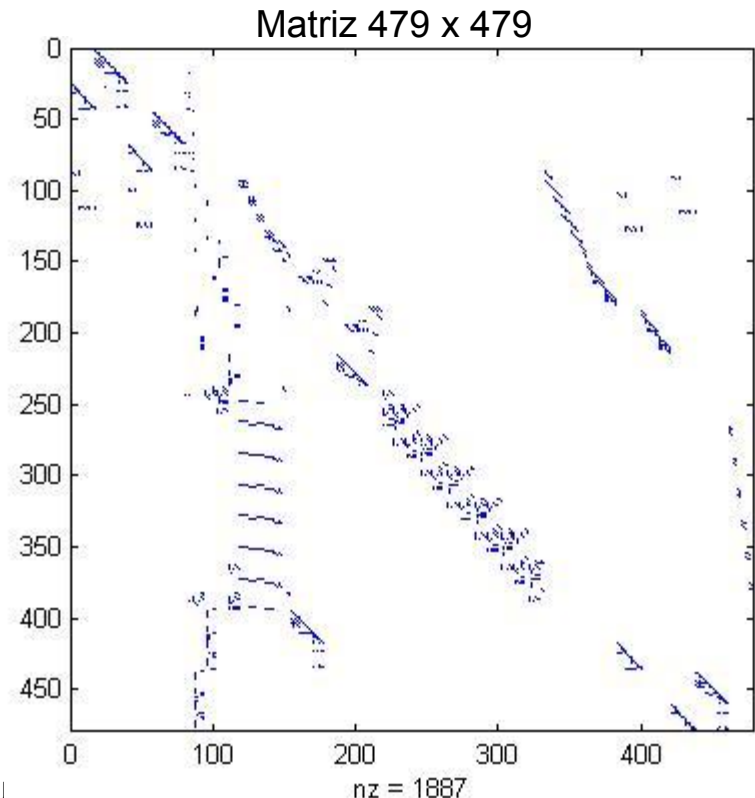
Almacenamiento compacto de matrices esparcidas (ralas)

Una matriz se denomina esparcida o rala cuando la mayoría de sus entradas son ceros.

Estas matrices aparecen en ciencia e ingeniería, por ejemplo, cuando se resuelve numéricamente ecuaciones diferenciales parciales.

Cuando se almacenan y manipulan este tipo de matrices en un computador, es beneficioso usar algoritmos y estructuras de datos especiales que tomen en cuenta la estructura esparcida de estas matrices.

Veremos 3 esquemas de almacenamiento para la matriz A.



Sistemas de Ecuaciones Lineales

Almacenamiento compacto de matrices esparcidas (ralas) – Esquema 1

La matriz A se almacena a través de tres arreglos XA , IF , IC , los cuales se describen a continuación:

XA : arreglo unidimensional real que contiene los elementos no nulos de la matriz, tomados fila por fila, comenzando desde la primera. Su dimensión es igual al número de elementos no nulos de la matriz.

IF : arreglo unidimensional entero que contiene los apuntadores al comienzo de cada fila de la matriz en A sobre el vector XA . Su dimensión es igual al orden de la matriz más uno.

IC : arreglo unidimensional entero que contiene los índices de las columnas de los elementos de la matriz en A . Su dimensión es igual a la de XA

Así, en la i -ésima fila de la matriz hay $IF(i+1)-IF(i)$ elementos no nulos, los cuales están almacenados consecutivamente en

$$XA(IF(i)), XA(IF(i)+1), \dots, XA(IF(i+1)-1),$$

y los índices de columna están almacenados consecutivamente en

$$IC(IF(i)), IC(IF(i)+1), \dots, IC(IF(i+1)-1).$$

Sistemas de Ecuaciones Lineales

Almacenamiento compacto de matrices esparcidas (ralas) – Esquema 1

$$A = \begin{pmatrix} 0.1 & 1.1 & 0 & 0 & -5.0 \\ 3.0 & 1.0 & 0 & 0 & 0 \\ 0 & 0 & 4.3 & 0 & 0 \\ 5.2 & 0 & 0 & -1.0 & 4.8 \\ 0 & 0 & 0.3 & 0 & -2.0 \end{pmatrix}$$

$$XA = (0.1 \quad 1.1 \quad -5.0 \quad 3.0 \quad 1.0 \quad 4.3 \quad 5.2 \quad -1.0 \quad 4.8 \quad 0.3 \quad -2.0)$$

$$IF = (1 \quad 4 \quad 6 \quad 7 \quad 10 \quad 12)$$

$$IC = (\underbrace{1 \quad 2 \quad 5}_1 \quad \underbrace{1 \quad 2}_2 \quad \underbrace{3}_3 \quad \underbrace{1 \quad 4 \quad 5}_4 \quad \underbrace{3 \quad 5}_5)$$

Sistemas de Ecuaciones Lineales

Almacenamiento compacto de matrices esparcidas (ralas) – Esquema 2

Para representar una matriz esparcida A de dimensión $n \times n$ usaremos el esquema denominado "indexado por filas" que contempla dos vectores de nombres sa e ija , donde sa almacena los elementos distintos de cero de A e ija almacena valores enteros correspondientes a apuntadores de filas y columnas para A .

Las reglas para el llenado de sa e ija son las siguientes (todos los vectores arrancan en 1 y no en 0):

Sistemas de Ecuaciones Lineales

Almacenamiento compacto de matrices esparcidas (ralas) – Esquema 2

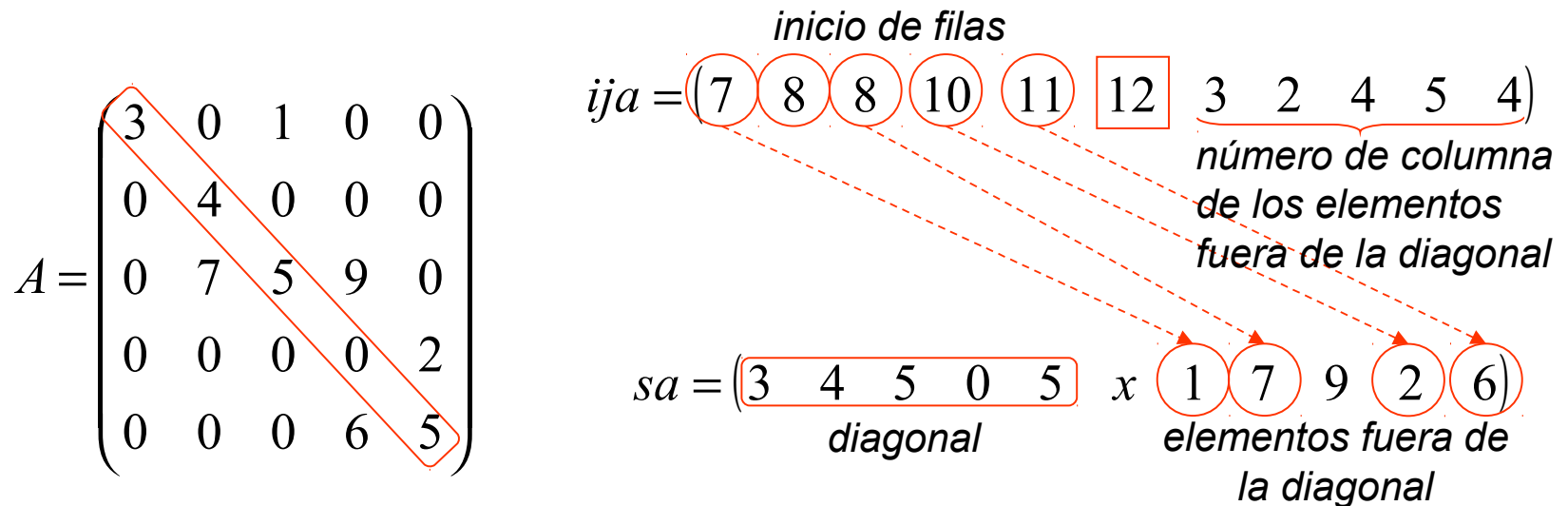
- En las primeras n posiciones de sa se almacenan los elementos de la diagonal de A (inclusive si los elementos son ceros).
- La posición $n+1$ en sa es arbitraria.
- Las entradas en sa correspondientes a las posiciones mayores o iguales a $n+2$ contienen las entradas no diagonales de A , ordenadas por fila, y dentro de cada fila ordenadas por columna.
- Cada una de las primeras n posiciones de ija almacena el índice sobre el vector sa donde comienza cada fila de A para los elementos fuera de la diagonal principal. Así $ija(k+1) - ija(k)$ nos da el número de elementos no diagonal distinto de cero de la fila k de la matriz A .
- La posición 1 sobre ija es siempre $n+2$. Esta información puede ser usada para conocer la dimensión de la matriz A .
- La posición $n+1$ sobre ija es 1 más que el índice en sa del último elemento no diagonal. Esta información puede ser usada para conocer el número de elementos no diagonales distintos de cero en A , es decir $ija(n+1) - 1 - (n+1)$.

Sistemas de Ecuaciones Lineales

Almacenamiento compacto de matrices esparcidas (ralas) – Esquema 2

- Las entradas en ija en las posiciones mayores o iguales a $n+2$ contienen los números de las columnas del elemento respectivo en sa .
- Los vectores sa e ija tienen la misma longitud, es decir $ija(ija(1)-1)-1$.

Ejemplo del llenado de los vectores sa e ija para la matriz A :



Así, $ija(1) = 7 = 5 + 2$ entonces $n = 5$, el número de elementos fuera de la diagonal distintos de cero es $ija(5+1)-1 - (5+1) = 5$, la dimensión de los vectores sa e ija es $ija(ija(1)-1)-1 = 11$.

Sistemas de Ecuaciones Lineales

Almacenamiento compacto de matrices esparcidas (ralas) – Esquema 3

Uno de estos formatos de almacenamiento es el denominado coordenado mediante el cual se almacenan los elementos no nulos de una matriz empleando tres vectores correspondientes a los índices de la fila, columna y a su valor correspondiente en la matriz.

Por ejemplo, dada la matriz A

$$A = \begin{pmatrix} 0.0 & -1.0 & 0.0 & 2.0 & 0.0 \\ 1.1 & 0.0 & -3.6 & 0.0 & 0.0 \\ 2.1 & 0.0 & 0.0 & 0.0 & 1.0 \\ 0.0 & 0.0 & 0.0 & -100.0 & -2.0 \\ -2.7 & 0.0 & 5.1 & 0.0 & -2.7 \end{pmatrix}$$

y denotando por $F(i)$, $C(i)$, $S(i)$, para $i = 1, \dots, n$, las componentes de los vectores fila (F), columna (C) y entrada no nula de la matriz (S), se tiene que el almacenamiento coordenado de A viene dado por:

Sistemas de Ecuaciones Lineales

Almacenamiento compacto de matrices esparcidas (ralas) – Esquema 3

$$F = (1 \ 1 \ 2 \ 2 \ 3 \ 3 \ 4 \ 4 \ 5 \ 5 \ 5)$$

$$C = (2 \ 4 \ 1 \ 3 \ 1 \ 5 \ 4 \ 5 \ 1 \ 3 \ 5)$$

$$S = (-1.0 \ 2.0 \ 1.1 \ -3.6 \ 2.1 \ 1.0 \ -100.0 \ -2.0 \ -2.7 \ 5.1 \ -2.7)$$

Es de hacer notar que la matriz A fue recorrida por filas para generar el almacenamiento coordinado.

Sistemas de Ecuaciones Lineales

Métodos iterativos - Resultados de convergencia

Dado el sistema lineal $AX = b$, se propone el método iterativo

$$MX^{(i+1)} = (M - A)x^{(i)} + b \quad (8)$$

donde M es una matriz invertible.

Teorema 1. Si $\delta = \|I - M^{-1}A\| < 1$, entonces la sucesión producida por el método (8) converge a la solución de $AX = b$, para cualquier vector inicial $x^{(0)}$.

Prueba:

Sea x solución de $AX = b$. Entonces x también es solución de

$$x = (I - M^{-1}A)x + M^{-1}b. \quad (9)$$

Ahora la iteración (8) es equivalente

$$x^{(k)} = (I - M^{-1}A)x^{(k-1)} + M^{-1}b. \quad (10)$$

Restando (9) de (10) obtenemos

Sistemas de Ecuaciones Lineales

Métodos iterativos - Resultados de convergencia

Prueba (cont.):

$$x^{(k)} - x = (I - M^{-1}A)(x^{(k-1)} - x).$$

de donde se tiene

$$\|x^{(k)} - x\| \leq \|(I - M^{-1}A)\| \|x^{(k-1)} - x\| = \delta \|x^{(k-1)} - x\| \quad (11)$$

Aplicando esta desigualdad en forma recursiva, resulta

$$\|x^{(k)} - x\| \leq \delta \|x^{(k-1)} - x\| \leq \delta^2 \|x^{(k-2)} - x\| \leq \dots \leq \delta^k \|x^{(0)} - x\| \quad (12)$$

para cualquier índice natural k . Tomando límite cuando $k \rightarrow \infty$ nos queda

$$0 \leq \lim_{k \rightarrow \infty} \|x^{(k)} - x\| \leq \lim_{k \rightarrow \infty} \delta^k \|x^{(0)} - x\| = 0$$

ya que $0 < \delta < 1$

Así, la sucesión del método iterativo converge a la solución x del sistema lineal $Ax = b$.



Sistemas de Ecuaciones Lineales

Métodos iterativos - Resultados de convergencia

Lema 1. Si $\delta = \|I - M^{-1}A\| < 1$, entonces la sucesión producida por el método (8) cumple:

$$\|x^{(k)} - x\| \leq \frac{\delta}{1 - \delta} \|x^{(k)} - x^{(k-1)}\|$$

Prueba: Como x es solución de $Ax = b$, también es solución de

$$x = (I - M^{-1}A)x + M^{-1}b.$$

Ahora la iteración (8) es equivalente

$$x^{(k)} = (I - M^{-1}A)x^{(k-1)} + M^{-1}b.$$

Restando las últimas ecuaciones se obtiene

$$x^{(k)} - x = (I - M^{-1}A)(x^{(k-1)} - x).$$

Así, tomando norma sigue

$$\|x^{(k)} - x\| \leq \|I - M^{-1}A\| \|x^{(k-1)} - x\| = \delta \|x^{(k-1)} - x\| \quad (13)$$

Sistemas de Ecuaciones Lineales

Métodos iterativos - Resultados de convergencia

Prueba (cont.)

Pero

$$\|x^{(k-1)} - x\| = \|x^{(k-1)} - x^{(k)} + x^{(k)} - x\| \leq \|x^{(k-1)} - x^{(k)}\| + \|x^{(k)} - x\|,$$

usando (13)

$$\|x^{(k-1)} - x\| \leq \|x^{(k-1)} - x^{(k)}\| + \delta \|x^{(k-1)} - x\|$$

$$\Leftrightarrow (1 - \delta) \|x^{(k-1)} - x\| \leq \|x^{(k-1)} - x^{(k)}\|$$

$$\Leftrightarrow \|x^{(k-1)} - x\| \leq \frac{1}{(1 - \delta)} \|x^{(k-1)} - x^{(k)}\|. \quad (14)$$

De (13) y (14)

$$\|x^{(k)} - x\| \leq \frac{\delta}{1 - \delta} \|x^{(k)} - x^{(k-1)}\|$$



Sistemas de Ecuaciones Lineales

Métodos iterativos - Resultados de convergencia

Teorema 2.

- Si A es diagonal dominante, entonces la sucesión producida por la iteración de Jacobi

$$x^{(i+1)} = D^{-1}(E + F)x^{(i)} + D^{-1}b$$

converge a la solución de $AX = b$ para cualquier vector inicial $x^{(0)}$.

- Si A es diagonal dominante, entonces la sucesión producida por la iteración de Gauss-Seidel

$$x^{(i+1)} = (D - E)^{-1}F x^{(i)} + (D - E)^{-1}b$$

converge a la solución de $AX = b$ para cualquier vector inicial $x^{(0)}$.

Obs. Para una matriz arbitraria A , la convergencia de uno de estos métodos no implica la convergencia del otro.

Sistemas de Ecuaciones Lineales

Métodos iterativos - Resultados de convergencia

Prueba. Caso iteración de Jacobi.

Primero
$$\|D^{-1}(E + F)\|_{\infty} = \max_{1 \leq i \leq n} \sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij} / a_{ii}|$$

porque para la matriz de $D^{-1}(E + F)$ se tiene

• elementos no diagonal $-a_{ij} / a_{ii}$

• elementos diagonal 0

$$= D^{-1}(E + F) = \begin{pmatrix} 0 & -\frac{a_{12}}{a_{11}} & \dots & \dots & -\frac{a_{1n}}{a_{11}} \\ -\frac{a_{21}}{a_{22}} & 0 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & 0 & -\frac{a_{n-1n}}{a_{n-1n-1}} \\ -\frac{a_{n1}}{a_{nn}} & \dots & \dots & -\frac{a_{nn-1}}{a_{nn}} & 0 \end{pmatrix}$$

Del hecho de que A es diagonal dominante se tiene

$$|a_{ii}| > \sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}| \quad \text{para } 1 \leq i \leq n \Leftrightarrow \sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij} / a_{ii}| < 1 \quad \text{para } 1 \leq i \leq n.$$

Entonces
$$\|D^{-1}(E + F)\|_{\infty} = \max_{1 \leq i \leq n} \sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij} / a_{ii}| < 1$$

$$\Leftrightarrow \|I - M^{-1}A\|_{\infty} = \|M^{-1}N\|_{\infty} < 1$$

y usando el teorema 1, se tiene que la iteración de Jacobi usando cualquier vector inicial $x^{(0)}$ converge a la solución del sistema $Ax = b$.



Sistemas de Ecuaciones Lineales

Métodos iterativos - Resultados de convergencia

Teorema 3. Consideremos el método iterativo

$$x^{(i+1)} = Hx^{(i)} + c \quad \text{con } H = M^{-1}N \quad \text{y } c = M^{-1}b.$$

el cual salió de descomponer $A = M - N$.

Para cualquier $x^{(0)} \in R^n$ el método iterativo converge si y sólo si $\rho(H) < 1$, (radio espectral de H es menor que 1).

Corolario. Si $\|H\| < 1$ entonces $x^{(i)}$ converge a x solución de $Ax=b$.

Prueba. Se deja como ejercicio.

Cotas de Error.

$$\text{a) } \|x^{(i)} - x\| \leq \|H\|^i \|x^{(0)} - x\|,$$

$$\text{b) } \|x^{(i)} - x\| \leq \frac{\|H\|^i}{1 - \|H\|} \|x^{(1)} - x^{(0)}\|, \quad \text{si } \|H\| < 1$$

Sistemas de Ecuaciones Lineales

Métodos iterativos - Resultados de convergencia

Obs.

La desigualdad en a) coincide con la ecuación (12) de la prueba del Teorema 1. Por lo tanto, la demostración de dicha desigualdad es exactamente igual a la deducción de la ecuación (12).

Obs.

En la desigualdad a), si se parte del vector inicial nulo, se obtiene una cota superior del error relativo de aproximar x por $x^{(i)}$:

$$\frac{\|x^{(i)} - x\|}{\|x\|} \leq \|H\|^i$$

Esto permite calcular aproximadamente el número de iteraciones necesarias para alcanzar una tolerancia dada.

Sistemas de Ecuaciones Lineales

Métodos iterativos - Resultados de convergencia

$$\text{Prueba de b) } \|x^{(i)} - x\| \leq \frac{\|H\|^i}{1 - \|H\|} \|x^{(1)} - x^{(0)}\|, \quad \text{si } \|H\| < 1$$

Como $\|H\| < 1$

$$\|x^{(0)} - x\| = \|x^{(0)} - x^{(1)} + x^{(1)} - x\| \leq \|x^{(0)} - x^{(1)}\| + \|x^{(1)} - x\|$$

Usando la desigualdad (a)

$$\begin{aligned} \|x^{(0)} - x\| &\leq \|x^{(0)} - x^{(1)}\| + \|H\| \|x^{(0)} - x\| \\ \Rightarrow (1 - \|H\|) \|x^{(0)} - x\| &\leq \|x^{(0)} - x^{(1)}\| \\ \Rightarrow \|x^{(0)} - x\| &\leq \frac{1}{(1 - \|H\|)} \|x^{(0)} - x^{(1)}\| \end{aligned}$$

Combinando con la desigualdad (a)

$$\|x^{(i)} - x\| \leq \|H\|^i \|x^{(0)} - x\| \leq \frac{\|H\|^i}{(1 - \|H\|)} \|x^{(0)} - x^{(1)}\|$$



Sistemas de Ecuaciones Lineales

Métodos iterativos - Resultados de convergencia

Obs.

En la desigualdad b) se obtiene una cota superior del error absoluto de aproximar x por $x^{(i)}$:

$$\|x^{(i)} - x\| \leq \frac{\|H\|^i}{1 - \|H\|} \|x^{(1)} - x^{(0)}\|$$

Esto permite calcular aproximadamente el número de iteraciones necesarias para alcanzar una tolerancia dada, para lo cual se debe calcular el iterado $x^{(1)}$.

Sistemas de Ecuaciones Lineales

Métodos iterativos - Resultados de convergencia

Comparación de los métodos de Jacobi y Gauss-Seidel

Teorema 4. Si A es diagonal dominante, entonces para cualquier vector inicial $x^{(0)} \in R^n$, los métodos de Jacobi y Gauss-Seidel convergen a la solución de $Ax = b$, y además se tiene

$$\|H_{GS}\|_{\infty} \leq \|H_J\|_{\infty} < 1$$

donde $H_{GS} = (D - E)^{-1}F$ y $H_J = D^{-1}(E + F)$ para $A = D - E - F$.

Teorema 5. Si $a_{ij} \leq 0$ si $i \neq j$ y $a_{ii} > 0$ si $1 \leq i \leq n$, entonces se satisface una y solamente una de las condiciones siguientes

- $0 < \rho(H_{GS}) < \rho(H_J) < 1$
- $1 < \rho(H_J) < \rho(H_{GS})$
- $\rho(H_{GS}) = \rho(H_J) = 1$
- $\rho(H_{GS}) = \rho(H_J) = 0$

Obs. O ambos métodos convergen o ambos divergen, y cuando convergen, Gauss-Seidel es más rápido de Jacobi, para este tipo de matrices.

Sistemas de Ecuaciones Lineales

Métodos iterativos - Resultados de convergencia

Teorema 6. Para $0 < \omega < 2$, si A es simétrica y definida positiva entonces el método de Relajación Sucesiva converge para cualquier $x^{(0)} \in \mathbb{R}^n$

Obs. El teorema 6 incluye al método de Gauss-Seidel ya que el método de Relajación Sucesiva coincide con Gauss-Seidel cuando $\omega = 1$

Obs. El recíproco del teorema 6 no es cierto.

$$A = \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}$$

$$\rho(H_{SOR}) = 0.8 < 1 \quad \text{para} \quad \omega = 0.2$$

el método de relajación sucesiva converge.

Se puede demostrar que A es definida positiva.

A no es simétrica.

Sistemas de Ecuaciones Lineales

Métodos iterativos - Resultados de convergencia

Obs. Si se agregan las condiciones A simétrica y $a_{ii} > 0$ para cada $i = 1, 2, \dots, n$ manteniendo $0 < \omega < 2$, entonces el recíproco del teorema 6 es cierto.

Es decir, bajo las hipótesis anteriores, partiendo de la convergencia del SOR para cualquier x_0 , se demuestra que A es definida positiva.

Obs. Aunque A sea simétrica y definida positiva, el método de Jacobi puede ser divergente.

Teorema 7. Si $a_{ii} \neq 0$ para cada $i = 1, 2, \dots, n$ entonces $\rho(H_{SOR}) \geq |\omega - 1|$. Así, la única manera de que $\rho(H_{SOR}) < 1$ es que $0 < \omega < 2$.

Sistemas de Ecuaciones Lineales

Métodos iterativos - Resultados de convergencia

Ejemplos:

$$A = \begin{pmatrix} -10 & 1 \\ 2 & -20 \end{pmatrix} \quad \begin{array}{l} \text{diagonal dominante (uso teorema 2)} \\ \text{Jacobi converge} \end{array}$$

$$A = \begin{pmatrix} -10 & 100 \\ 2 & -20 \end{pmatrix} \quad \begin{array}{l} \rho(H_J) > 1 \quad \text{no diagonal dominante} \\ \text{Jacobi diverge} \\ \text{(teorema 3)} \end{array}$$

$$A = \begin{pmatrix} -10 & 100 \\ 1 & -20 \end{pmatrix} \quad \begin{array}{l} \rho(H_J) < 1 \quad \text{no diagonal dominante} \\ \text{Jacobi converge} \\ \text{(teorema 3)} \end{array}$$

$$H_J = D^{-1}(E + F) = I - D^{-1}A$$

Sistemas de Ecuaciones Lineales

Métodos iterativos – Ejemplo de comparación

Para $s \in R$, considere la matriz simétrica

$$A = \begin{pmatrix} 1 & s & s \\ s & 1 & s \\ s & s & 1 \end{pmatrix}$$

Cálculos con MATLAB:
 $M = \text{sym}([1 \ s \ s; \ s \ 1 \ s; \ s \ s \ 1])$
 $\text{determ} = \text{det}(M)$
 $\text{pc} = \text{poly}(M)$
 $\text{factores} = \text{factor}(\text{pc})$

Los valores propios de A son: $1 - s$ (con multiplicidad 2) y $1 + 2s$.

La matriz A es definida positiva cuando $s \in [-0.5, 1]$ y diagonal dominante estricta para $s \in [-0.5, 0.5]$

Resolviendo el sistema $Ax = b$ para diferentes valores de s , usando los métodos de Jacobi y Gauss-Seidel con $b = (1, 1, 1)^t$, $x^{(0)} = (0.5, 0.5, 0.5)^t$, y $n = 500$

Para ambos métodos se implementó el criterio de detención visto en clase, con una tolerancia de 10^{-8} ,

$$\|x^{(k)} - x^{(k-1)}\| < \text{tol}$$

Sistemas de Ecuaciones Lineales

Métodos iterativos – Ejemplo de comparación

- Para $s = 0.3$, la matriz A es positiva definida y diagonal dominante.

Jacobi itera 35 veces y Gauss-Seidel 11 veces.

Ambos métodos entregan como solución $x = (0.6250, 0.6250, 0.6250)^t$ que es la solución exacta. El radio espectral de H_J es 0.60 y H_{GS} es 0.1643 (ambos menores que 1).

- Para $s = 0.6$, la matriz A es positiva definida, pero no es diagonal dominante.

Jacobi no converge después de 500 iteraciones y Gauss-Seidel si converge en 22 iteraciones.

Gauss-Seidel entrega como solución $x = (0.45455, 0.45455, 0.45455)^t$ que es la solución exacta. El radio espectral de H_J es 1.20 y H_{GS} es 0.4648.

- Para $s = 1.01$, la matriz A no es positiva definida ni diagonal dominante.

Jacobi y Gauss-Seidel no convergen después de 500 iteraciones.

La solución exacta es $x = (0.3125, 0.3125, 0.3125)$. El radio espectral de H_J es 2.20 y H_{GS} es 1.1537 (ambos mayores que 1).

Sistemas de Ecuaciones Lineales

Métodos iterativos – Ejemplo de comparación

```

1 function [x,J,c] = jacobi(A,b,n,z,tol)
2 %
3 %   x = jacobi(A,b,n,z)
4 %
5 %   Jacobi iteration on system A*x = b with printing
6 %   n -- number of iterations
7 %   z -- initial vector   (default 0)
8 %   tol -- tolerance
9 %
10 %   x -- final iterate
11 %   J -- Jacobi matrix
12 %   c -- Jacobi vector
13 %
14
15 if nargin <=3
16     z=0*b;
17     tol = eps
18 end
19
20 D = diag(diag(A));
21 J = D\(D - A);
22 c = D\b;
23 x=z;
24 for k = 1:n
25     y = J*x + c;
26     fprintf(1,'%3d      ',k)
27     fprintf(1,'%5.4f      ',y')
28     fprintf(1,'\n')
29     if norm(x-y)<tol
30         x = y;
31         return;
32     else
33         x = y;
34     end
35 end

```

```

1 function [x,G,c] = gsmv(A,b,n,z,tol)
2 %
3 %   x = gsmv(A,b,n,z)
4 %
5 %   Gauss-Seidel iteration on system A*x = b with printing
6 %   using matrix multiplication (not optimal)
7 %   n -- number of iterations
8 %   z -- initial vector   (default 0)
9 %   tol -- tolerance
10 %
11 %   x -- final iterate
12 %   G -- Gauss-Seidel matrix
13 %   c -- Gauss-Seidel vector
14 %
15
16 if nargin <=3
17     z=0*b;
18     tol = eps
19 end
20
21 LD = tril(A);
22 G = -LD\triu(A,1);
23 c = LD\b;
24 x=z;
25 for i = 1:n
26     y = G*x + c;
27     fprintf(1,'%3d      ',i)
28     fprintf(1,'%5.5f      ',y')
29     fprintf(1,'\n')
30     if norm(x-y)<tol
31         x = y;
32         return;
33     else
34         x = y;
35     end
36 end

```



Sistemas de Ecuaciones Lineales

Métodos directos (MD) vs iterativos (MI)

- MD no requieren estimado inicial; MI aprovechan conocimiento de un buen estimado
- MD dan alta precisión; MI aprovechan el caso en que no se necesita alta precisión
- MI dependen de propiedades estructurales y la convergencia es lenta para sistemas mal condicionados; MD son más robustos
- MI requieren menos trabajo (si la convergencia es rápida) pero a menudo requieren preconditionamiento
- MI no requieren almacenamiento explícito de la matriz del sistema
- MD son más estándar y se consiguen más fácilmente en paquetes de software numérico

Sistemas de Ecuaciones Lineales

Autovalores y Autovectores

El espacio vectorial C^n consiste de todos las n-tuplas complejas (vectores complejos) de la forma

$$x = (x_1, \dots, x_n)^T \quad \text{donde } x_j \in C \quad \text{para } 1 \leq j \leq n.$$

Para $\lambda \in C$, $\lambda x = (\lambda x_1, \dots, \lambda x_n)^T$.

C^n es un espacio vectorial sobre el campo escalar C .

En C^n definimos el producto interno y norma euclidea respectivamente por

$$\langle x, y \rangle = \sum_{j=1}^n x_j \overline{y_j} \quad \text{y} \quad \|x\|_2 = \sqrt{\langle x, x \rangle}$$

Obs. Para $x, y, z \in C^n$ y $\lambda \in C$

$$a) \langle x, y \rangle = \overline{\langle y, x \rangle}$$

$$b) \langle x, \lambda y \rangle = \overline{\lambda} \langle x, y \rangle$$

$$c) \langle x + y, z \rangle = \langle x, z \rangle + \langle y, z \rangle$$

Sistemas de Ecuaciones Lineales

Autovalores y Autovectores

Obs. En \mathbb{C} , se tiene el teorema fundamental del algebra

“Todo polinomio no constante con coeficientes complejos tiene al menos un cero en \mathbb{C} ”.

De aquí,

“Todo polinomio de grado n puede expresarse como un producto de n factores lineales”.

Definición. Si A es una matriz con elementos en \mathbb{C} , la transpuesta conjugada de A , se denota por A^* y es $(a_{ij})^* = (\overline{a_{ji}})$.

Si x es una matriz $n \times 1$
(vector columna),
entonces

$$\left\{ \begin{array}{l} a) x^* = (\overline{x_j})_{1 \leq j \leq n} \\ b) y^* x = \langle x, y \rangle = \sum_{j=1}^n x_j \overline{y_j} \\ c) x^* x = \langle x, x \rangle = \|x\|_2^2 = \sum_{j=1}^n x_j \overline{x_j} = \sum_{j=1}^n |x_j|^2 \end{array} \right.$$

Sistemas de Ecuaciones Lineales

Autovalores y Autovectores

Sea A una matriz $n \times n$, $a_{ij} \in \mathbb{C}$, sea $\lambda \in \mathbb{C}$.

Si la ecuación

$$Ax = \lambda x \quad (1)$$

tiene una solución no trivial, es decir, $x \neq 0$, entonces λ se denomina un **autovalor** de A . El vector x no cero que satisface (1), se denomina **autovector** de A correspondiente al autovalor λ .

Ejemplo:

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

-2 es un autovalor de la matriz 3×3 dada, y el vector $(1, 3, -4)^T$ es un autovector correspondiente.

Sistemas de Ecuaciones Lineales

Autovalores y Autovectores

La condición de que (1) tiene una solución no trivial es equivalente a

- $A - \lambda I$ mapea vectores no cero en el vector nulo (2)

- $A - \lambda I$ es singular (3)

- $\det(A - \lambda I) = 0$ (4)

La ecuación (4) se conoce como la **ecuación característica** de la matriz A .

El lado izquierdo de (4) es un polinomio de grado n en la variable λ y se denomina **polinomio característico** de A .

Obs. Toda matriz $n \times n$ tiene exactamente n autovalores, incluyendo aquí todos las posibles multiplicidades que estos poseen como raíces de la ecuación característica.

Obs. En matrices pequeñas, los autovalores pueden ser calculados resolviendo en λ la ecuación (4). Para matrices grandes, este método no se recomienda. Una razón es que las raíces del polinomio pueden ser sensitivas como función de los coeficientes del polinomio.

Sistemas de Ecuaciones Lineales

Autovalores y Autovectores

Ejemplo:

$$A = \begin{bmatrix} -1 & 10 \\ 0 & -2 \end{bmatrix}$$

$$\det(\lambda I - A) = 0 \Rightarrow \lambda_1 = -1, \lambda_2 = -2$$

$$\begin{bmatrix} -1 & 10 \\ 0 & -2 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = -1 \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \Rightarrow \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} -1 & 10 \\ 0 & -2 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = -2 \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \Rightarrow \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} -10 \\ 1 \end{bmatrix}$$

Cálculos con MATLAB:
`M = sym('[-1,10;0,-2]')`
`determ = det(M)`
`pc = poly(M)`
`factores = factor(pc)`

Sistemas de Ecuaciones Lineales

Localizando autovalores.

Teorema de Gerschgorin.

El espectro de una matriz A de orden n (conjunto de todos los autovalores de A) está contenido en la unión de los discos D_i , $i=1, \dots, n$ en el plano complejo, donde

$$D_i = \left\{ z \in \mathbb{C} : |z - a_{ii}| \leq \sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}| \right\} \quad \text{para } 1 \leq i \leq n$$

Además, la unión de cualesquiera k de estos discos que no interseccione a los $(n-k)$ restantes, debe contener exactamente k autovalores (contando multiplicidades)

Prueba.

λ autovalor de A y sea x el autovector asociado con $\|x\|_\infty = 1$, entonces $Ax = \lambda x$ y existe i tal que $|x_i| = 1$. Como $(Ax)_i = \lambda x_i$ sigue

$$\sum_{j=1}^n a_{ij} x_j = \lambda x_i \Leftrightarrow a_{ii} x_i + \sum_{\substack{j=1 \\ j \neq i}}^n a_{ij} x_j = \lambda x_i \Leftrightarrow (\lambda - a_{ii}) x_i = \sum_{\substack{j=1 \\ j \neq i}}^n a_{ij} x_j$$

Sistemas de Ecuaciones Lineales

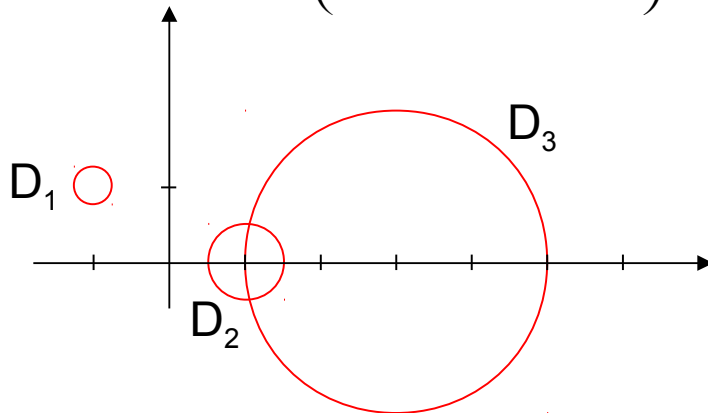
Localizando autovalores (cont.).

Tomado módulo, aplicando la desigualdad triangular y usando $|x_j| \leq 1 = |x_i|$ se tiene

$$|\lambda - a_{ii}| \leq \sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}| |x_j| \leq \sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}|$$

es decir, $\lambda \in D_i$.

Ejemplo. Si $A = \begin{pmatrix} -1+i & 0 & 1/4 \\ 1/4 & 1 & 1/4 \\ 1 & 1 & 3 \end{pmatrix}$ se tiene $\begin{cases} D_1 = \{z \in \mathbb{C} : |z - (-1+i)| \leq 1/4\} \\ D_2 = \{z \in \mathbb{C} : |z - 1| \leq 1/2\} \\ D_3 = \{z \in \mathbb{C} : |z - 3| \leq 2\} \end{cases}$



$$\lambda_1 = -1.0540 + 0.9888i$$

$$\lambda_3 = 3.1780 + 0.0141i$$

$$\lambda_2 = 0.8761 - 0.0030i$$

$$A = [-1+i, 0, 1/4; 1/4, 1, 1/4; 1, 1, 3]$$

Sistemas de Ecuaciones Lineales

Localizando autovalores (cont.).

¿Existe alguna relación entre los autovalores de una matriz y los de su traspuesta?

¿Qué se puede comentar acerca de los círculos de A y de su traspuesta?

Sistemas de Ecuaciones Lineales

Método de la Potencia

Está diseñado para calcular el autovalor dominante y el autovector correspondiente. (2)

Suposiciones: A una matriz $n \times n$, para la cual

- a) existe un autovalor simple de módulo máximo y
- b) hay independencia lineal del conjunto de n autovectores.

Según (a), los autovalores $\lambda_1, \dots, \lambda_n$ pueden ser reordenados tal que

$$|\lambda_1| > |\lambda_2| \geq \dots \geq |\lambda_n|.$$

Según (b), existe una base $\{u^{(1)}, \dots, u^{(n)}\}$ para C^n tal que

$$Au^{(j)} = \lambda_j u^{(j)} \quad \text{para } 1 \leq j \leq n. \quad (5)$$

Si $x^{(0)} \in C^n$ entonces

$$x^{(0)} = a_1 u^{(1)} + \dots + a_n u^{(n)} \quad \text{con } a_1 \neq 0. \quad (6)$$

Sistemas de Ecuaciones Lineales

Método de la Potencia

Construimos la sucesión

$$x^{(1)} = Ax^{(0)}, \quad x^{(2)} = Ax^{(1)}, \dots, \quad x^{(k)} = Ax^{(k-1)}.$$

Entonces

$$x^{(k)} = A^k x^{(0)}. \quad (7)$$

Podemos suponer en (6) sin pérdida de generalidad que

$$x^{(0)} = u^{(1)} + \dots + u^{(n)}, \quad (8)$$

es decir, los coeficientes a_j son absorbidos por los vectores $u^{(j)}$.

Sustituyendo (8) en (7) se tiene

$$x^{(k)} = A^k u^{(1)} + \dots + A^k u^{(n)},$$

y usando (5)

$$x^{(k)} = \lambda_1^k u^{(1)} + \dots + \lambda_n^k u^{(n)}$$

factorizando

$$= \lambda_1^k \left[u^{(1)} + \left(\frac{\lambda_2}{\lambda_1} \right)^k u^{(2)} \dots + \left(\frac{\lambda_n}{\lambda_1} \right)^k u^{(n)} \right]. \quad (9)$$

Sistemas de Ecuaciones Lineales

Método de la Potencia

Como $|\lambda_1| > |\lambda_j|$ para $2 \leq j \leq n$, se tiene

$$\left| \frac{\lambda_j}{\lambda_1} \right| < 1 \text{ para } 2 \leq j \leq n \text{ y } \left| \frac{\lambda_j}{\lambda_1} \right|^k \text{ tiende a cero cuando } k \rightarrow \infty.$$

Así, podemos escribir (9) como

$$x^{(k)} = \lambda_1^k [u^{(1)} + \varepsilon^{(k)}], \quad (10)$$

donde $\varepsilon^{(k)} \rightarrow 0$ cuando $k \rightarrow \infty$.

Sea φ una funcional lineal sobre C^n para el cual se satisface $\varphi(u^{(1)}) \neq 0$.

$(\varphi : C^n \rightarrow C, \varphi$ es lineal si

$$\varphi(\alpha x + \beta y) = \alpha \varphi(x) + \beta \varphi(y) \text{ para } \alpha, \beta \in C \text{ y } x, y \in C^n)$$

Sistemas de Ecuaciones Lineales

Método de la Potencia

Entonces, de (10)

$$\varphi(\mathbf{x}^{(k)}) = \lambda_1^k [\varphi(\mathbf{u}^{(1)}) + \varphi(\boldsymbol{\varepsilon}^{(k)})], \quad (11)$$

de donde, para $k \rightarrow \infty$ tomamos

$$r_k = \frac{\varphi(\mathbf{x}^{(k+1)})}{\varphi(\mathbf{x}^{(k)})} = \lambda_1 \frac{[\varphi(\mathbf{u}^{(1)}) + \varphi(\boldsymbol{\varepsilon}^{(k+1)})]}{[\varphi(\mathbf{u}^{(1)}) + \varphi(\boldsymbol{\varepsilon}^{(k)})]} \rightarrow \lambda_1.$$

Esto constituye el método de la potencia para calcular λ_1 .

Obs.

Como la dirección del vector $\mathbf{x}^{(k)}$ tiende a la dirección de $\mathbf{u}^{(1)}$ cuando $k \rightarrow \infty$ (usando (10)), el método nos permite calcular el autovector $\mathbf{u}^{(1)}$.

Sistemas de Ecuaciones Lineales

Método de la Potencia

Algoritmo de la
potencia

Leer $A=(a_{ij}), n, x, itmax$

Para $k = 1$ hasta $itmax$

$$y = Ax$$

$$r = \varphi(y) / \varphi(x)$$

$$x = y / \|y\|$$

Escribir k, x, r

Fin para

Obs.

- La normalización del vector x se introduce aquí, para evitar que converja a cero o la sucesión de vectores x deje de estar acotado.
- Al final r es el autovalor mayor y x un autovector correspondiente unitario.
- Como funcional lineal φ podemos tomar la proyección sobre la componente j : $\varphi : \mathbb{C}^n \rightarrow \mathbb{C}, \varphi(x) = x_j$.
- Posible criterio de parada en el paso k : $\|x^{(k-1)} - y / \|y\|_\infty\|_\infty < \varepsilon$.

Sistemas de Ecuaciones Lineales

Método de la Potencia

Ejemplo. Calcular el autovalor mayor y un autovector correspondiente para A

$$A = \begin{pmatrix} 6 & 5 & -5 \\ 2 & 6 & -2 \\ 2 & 5 & -1 \end{pmatrix} \quad Ax = \lambda x$$

$$x = (-1, 1, 1)^T \quad \text{vector inicial}$$

$$\varphi: \mathbb{C}^n \rightarrow \mathbb{C}, \quad \varphi(x) = x_2$$

$$A = [6,5,-5; 2,6,-2; 2,5,-1]$$

$$k = 1, r^{(1)} = 2.0, x^{(1)} = (-1.00000 \quad 0.33333 \quad 0.33333)$$

$$k = 2, r^{(2)} = -2.0, x^{(2)} = (-1.00000 \quad -0.11111 \quad -0.11111)$$

$$k = 3, r^{(3)} = 22.0, x^{(3)} = (-1.00000 \quad -0.40741 \quad 0.40741)$$

$$k = 4, r^{(4)} = 8.9091, x^{(4)} = (-1.00000 \quad -0.60494 \quad -0.60494)$$

...

$$k = 28, r^{(28)} = 6.00007, x^{(28)} = (-1.00000 \quad -0.99998 \quad -0.99998)$$

El mayor autovalor de A es 6 y su autovector es $(-1, -1, -1)^T$.

Sistemas de Ecuaciones Lineales

Método de la Potencia

Ejemplo. Calcular el autovalor mayor y un autovector correspondiente para A

$$A = [1.5, 0.5; 0.5, 1.5]$$

$$A = \begin{bmatrix} 1.5 & 0.5 \\ 0.5 & 1.5 \end{bmatrix} \quad y \quad x_0 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$y^{(1)} = Ax^{(0)} = [0.5, 1.5]^T$$

$$x^{(1)} = y^{(1)} / \|y^{(1)}\|_\infty = [0.333, 1.000]^T$$

$$x^{(2)} = [0.600, 1.000]^T$$

$$x^{(3)} = [0.778, 1.000]^T$$

⋮

$$x^{(10)} = [0.998, 1.000]^T$$

$$x^{(11)} = [0.999, 1.000]^T$$

$$\|y^{(11)}\|_\infty = 1.999 \Rightarrow |\lambda_1| \approx 1.999$$

$$u_1 \approx [0.999, 1.000]^T$$

$$Au_1 - \lambda_1 u_1 = \begin{bmatrix} 0.0005 \\ -0.0005 \end{bmatrix}$$

Sistemas de Ecuaciones Lineales

Método de la Potencia

Obs. El algoritmo de la potencia visto presenta el inconveniente en la escogencia de la funcional lineal φ . Una manera de independizar el algoritmo en la escogencia de φ es como sigue:

Partiendo de (10) se tiene que para k grande, $\lim_{k \rightarrow \infty} \frac{x^{(k)}}{\lambda_1^k} = u^{(1)}$,
de donde

$$Ax^{(k)} = x^{(k+1)} \approx \lambda_1^{k+1} u^{(1)} = \lambda_1 (\lambda_1^k u^{(1)}) \approx \lambda_1 x^{(k)}$$

Es decir, para k grande, $Ax^{(k)} \approx \lambda_1 x^{(k)}$

lo cual significa que en el limite $x^{(k)}$ es un autovector asociado a λ_1

Además, normalizando en norma 2 al vector $x^{(k)}$ se obtiene un autovector unitario (esto garantiza que la sucesión generada por este método está acotada).

Al detener el algoritmo en un valor de k determinado por el criterio de convergencia, se calcula el autovalor dominante λ_1 como sigue:

$$\langle Ax^{(k)}, x^{(k)} \rangle \approx \langle \lambda_1 x^{(k)}, x^{(k)} \rangle = \lambda_1 \langle x^{(k)}, x^{(k)} \rangle$$

Sistemas de Ecuaciones Lineales

Método de la Potencia

Obs (cont.).

de donde

$$\lambda_1 = \frac{\langle Ax^{(k)}, x^{(k)} \rangle}{\langle x^{(k)}, x^{(k)} \rangle} = \langle Ax^{(k)}, x^{(k)} \rangle$$

La nueva versión del algoritmo de la potencia es la siguiente:

Leer $A=(a_{ij})$, x , $itmax$, tol

Para $k = 1$ hasta $itmax$

$$y = x$$

$$x = Ax$$

si $\|x\|_2 = 0$, el método no converge

$$x = x / \|x\|_2$$

si $\|y - x\|_2 < tol$, $\lambda = \langle Ax, x \rangle$

Fin para

Sistemas de Ecuaciones Lineales

Método de la Potencia

Teorema. Si λ es un autovalor de A y si A es no singular, entonces λ^{-1} es un autovalor de A^{-1} .

Prueba.

Se tiene que $AX = \lambda X$ con $X \neq 0$.

Entonces $X = A^{-1}(\lambda X) = \lambda A^{-1}X$, como $\lambda \neq 0$ sigue $A^{-1}X = \lambda^{-1}X$, de donde λ^{-1} es un autovalor de A^{-1} .

Obs.

El teorema anterior sugiere una manera de calcular el autovalor más pequeño de A . Supongamos que los autovalores de A satisfacen

$$|\lambda_1| \geq |\lambda_2| \geq \dots \geq |\lambda_{n-1}| > |\lambda_n| > 0.$$

Esto garantiza que A es no singular, ya que 0 no es un autovalor de A .

Tenemos que los autovalores de A^{-1} son los números λ_j^{-1} , y estos se pueden reordenar así

$$|\lambda_n^{-1}| > |\lambda_{n-1}^{-1}| \geq \dots \geq |\lambda_2^{-1}| \geq |\lambda_1^{-1}| > 0.$$

Sistemas de Ecuaciones Lineales

Método de la Potencia

Obs. (cont.)

Ahora podemos aplicar el método de la potencia a A^{-1} para calcular su autovalor más grande λ_n^{-1} , es decir, hemos calculado λ_n el autovalor más pequeño de A .

En este caso no es buena idea calcular la inversa de A , es decir A^{-1} , para luego calcular $X^{(k+1)}$ usando la iteración

$$X^{(k+1)} = A^{-1}X^{(k)}.$$

En lugar de esto, procedemos así:

Obtenemos $X^{(k+1)}$ resolviendo el sistema $AX^{(k+1)} = X^{(k)}$,

mediante el método de descomposición LU (esto se lleva a cabo una sola vez), seguido por la resolución de 2 sistemas triangulares, donde el vector de la derecha cambia en cada iteración.

Este procedimiento se conoce como el **método de la potencia inverso**.

Sistemas de Ecuaciones Lineales

Método de la Potencia

Algoritmo de la
potencia inverso

Leer $A=(a_{ij}), n, x, itmax$

Calcular L y U tal que $A = LU$

Para $k = 1$ hasta $itmax$

resolver $LUy = x$

$r = \varphi(y) / \varphi(x)$

$x = y / \|y\|$

Escribir k, x, r

Fin para

Al final, r es el mayor autovalor para A^{-1} , de donde, usando el teorema, $1/r$ es el menor autovalor para A y x es un autovector asociado.

Sistemas de Ecuaciones Lineales

Método de la Potencia

Ejemplo. Calcular el autovalor menor y un autovector correspondiente para A

$$A = \begin{pmatrix} 6 & 5 & -5 \\ 2 & 6 & -2 \\ 2 & 5 & -1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 1/3 & 1 & 0 \\ 1/3 & 10/13 & 1 \end{pmatrix} \begin{pmatrix} 6 & 5 & -5 \\ 0 & 13/3 & -1/3 \\ 0 & 0 & 12/13 \end{pmatrix} \quad Ax = \lambda x$$

$x = (3 \ 7 \ -13)^T$ vector inicial

$\varphi: \mathbb{C}^n \rightarrow \mathbb{C}, \quad \varphi(x) = x_1$

En cada paso del algoritmo se calcula $x^{(k+1)}$ a partir de $LUx^{(k+1)} = x^{(k)}$.

A continuación se calcula en $r^{(k+1)} = x_1^{(k+1)} / x_1^{(k)}$.

Antes de continuar con el siguiente iterado, se normaliza $x^{(k+1)}$

dividiendo entre su norma infinito.

$$k = 1, \quad r^{(1)} = -5.8889, \quad x^{(1)} = (-0.80165 \quad -0.00826 \quad -1.00000)$$

$$k = 2, \quad r^{(2)} = 1.19759, \quad x^{(2)} = (-0.95089 \quad -0.01774 \quad -1.00000)$$

$$k = 3, \quad r^{(3)} = 1.02750, \quad x^{(3)} = (-0.98759 \quad -0.00712 \quad -1.00000)$$

$$\dots$$

$$k = 6, \quad r^{(6)} = 1.00012, \quad x^{(6)} = (-0.99980 \quad -0.00017 \quad -1.00000)$$

$$\dots$$

$$k = 11, \quad r^{(11)} = 1.00000, \quad x^{(11)} = (-1.00000 \quad 0.00000 \quad -1.00000)$$

El menor autovalor de A es 1 y un autovector es $(-1, 0, -1)^T$.

Sistemas de Ecuaciones Lineales

Método de la Potencia

Ejemplo. Calcular el autovalor menor y el autovector correspondiente para A

$$A = \begin{bmatrix} 1.5 & 0.5 \\ 0.5 & 1.5 \end{bmatrix} \quad y \quad x_0 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$\begin{aligned} & \vdots \\ x^{(10)} &= [-0.998, 1.000]^T \\ x^{(11)} &= [-0.999, 1.000]^T \end{aligned}$$

$$\begin{aligned} \|y^{(11)}\|_\infty &= 1.000 \Rightarrow |\lambda_n| \approx 1 \\ u_n &\approx [-0.999, 1.000]^T \end{aligned}$$

$$Au_n - \lambda_n u_n = \begin{bmatrix} 0.0005 \\ 0.0005 \end{bmatrix}$$

Sistemas de Ecuaciones Lineales

Método de la Potencia

Hasta el momento hemos cubierto

- el método de la potencia para calcular el autovalor dominante de una matriz A y
- el método de la potencia inverso para calcular el autovalor más pequeño en módulo de A .

Consideremos la matriz desplazada $A - \mu I$, de aquí podemos generar un procedimiento para calcular el autovalor de A más cercano a un valor dado μ .

Supongamos que un autovalor de A , digamos λ_i , satisface la desigualdad

$$0 < |\lambda_i - \mu| < \varepsilon$$

donde μ es un número complejo dado y $\varepsilon > 0$.

Supongamos que los otros autovalores de A satisfacen la desigualdad

$$|\lambda_j - \mu| > \varepsilon \quad \text{para } j \neq i.$$

Sistemas de Ecuaciones Lineales

Método de la Potencia

Como los autovalores de $A - \mu I$ son los números de la forma (probarlo) $\lambda_i - \mu$, aplicando el método de la potencia inverso a $A - \mu I$, se puede aproximar el autovalor dominante de $(A - \mu I)^{-1}$ que es de la forma

$$\xi_k = (\lambda_k - \mu)^{-1}.$$

Aquí, el autovector asociado se obtiene resolviendo la ecuación

$$(A - \mu I)x^{(k+1)} = x^{(k)},$$

donde se usa el método de descomposición LU (este se aplica una sola vez en este algoritmo).

Como el procedimiento calcula

$$\xi_k = (\lambda_k - \mu)^{-1},$$

el λ_k (autovalor mas cercano a μ) puede ser recuperado despejando

$$\lambda_k = \xi_k^{-1} + \mu.$$

Este procedimiento se conoce como el **método de la potencia inverso desplazado**.

Sistemas de Ecuaciones Lineales

Método de la Potencia

De manera similar, podemos calcular el autovalor, digamos λ_k , mas lejano de un valor dado μ .

Supongamos que existe $\varepsilon > 0$ tal que

$$|\lambda_k - \mu| > \varepsilon$$

para un autovalor λ_k de A, y para los otros autovalores $\lambda_j, j \neq k$, se tiene

$$0 < |\lambda_j - \mu| < \varepsilon \quad j \neq k.$$

Usando el método de la potencia aplicado a $A - \mu I$ (autovalores $\lambda_k - \mu$), podemos calcular su autovalor dominante

$$\xi_k = \lambda_k - \mu,$$

de donde podemos recuperar λ_k , el autovalor mas lejano, como

$$\lambda_k = \xi_k + \mu.$$

Este procedimiento se conoce como el **método de la potencia desplazado**.

Sistemas de Ecuaciones Lineales

Método de la Potencia

Resumen. Supongamos que los autovalores de A satisfacen

$$|\lambda_1| > |\lambda_2| \geq \dots \geq |\lambda_{n-1}| > |\lambda_n| > 0.$$

Método	Ecuación	Objetivo
potencia	$x^{(k+1)} = Ax^{(k)}$	autovalor dominante λ_1
potencia inverso	$Ax^{(k+1)} = x^{(k)}$	autovalor más pequeño λ_n
potencia desplazado	$x^{(k+1)} = (A - \mu I)x^{(k)}$	autovalor más lejano a μ
potencia inverso desplazado	$(A - \mu I)x^{(k+1)} = x^{(k)}$	autovalor más cercano a μ

Sistemas de Ecuaciones Lineales

Método de la Potencia

Ejercicio.

Dados una matriz A de dimensión $n \times n$, μ un número complejo, escribir procedimientos para MATLAB que permita calcular el autovalor más cercano y más alejado de μ , usando los métodos de la potencia inverso desplazado y de la potencia desplazado. Aplicarlo a la matriz A dada para calcular el autovalor más cercano a 3.

$$A = \begin{pmatrix} 6 & 5 & -5 \\ 2 & 6 & -2 \\ 2 & 5 & -1 \end{pmatrix}$$

Autovalores: 6, 4 y 1

potencia
inverso
desplazado

$$(A - \mu I) x^{(k+1)} = x^{(k)}$$

$$x^{(0)} = (1, 1/2, 1)^T$$

$$(A - 3I) = \begin{pmatrix} 3 & 5 & -5 \\ 2 & 3 & -2 \\ 2 & 5 & -4 \end{pmatrix}$$

$$A - 3I = [3, 5, -5; 2, 3, -2; 2, 5, -4]$$

Sistemas de Ecuaciones Lineales

Método de la Potencia. Localizando autovalores.

Ejemplo. Calcular el autovalor que tiene parte real más negativa

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

Utilizando los círculos de Gerschgorin, se deduce que el autovalor de parte real más negativa será aquel más cercano a $\sigma = -4$.

Aplicar método a $A + 4I$:

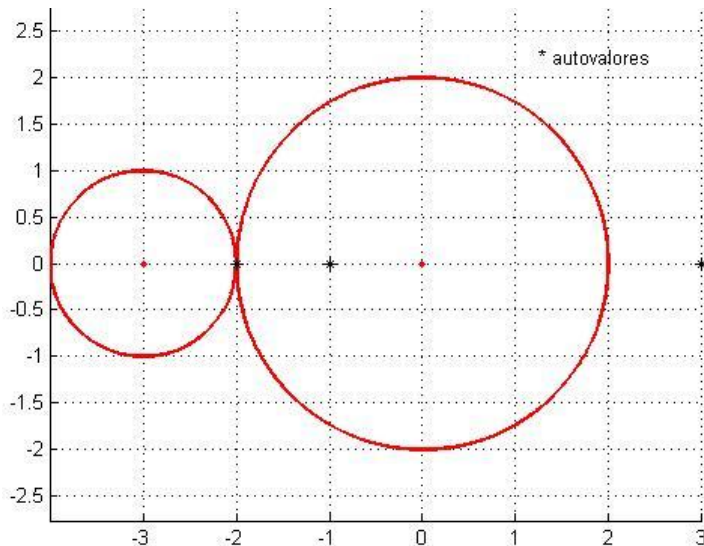
$$x^{(0)} = (1, 1, 1)^T$$

$$\begin{aligned} & \vdots \\ x^{(6)} &= [1.0000, 1.000, 0.0005]^T \\ x^{(7)} &= [1.0000, 1.0000, 0.0001]^T \end{aligned}$$

$$\|y^{(7)}\|_{\infty} = 0.5 \Rightarrow |\lambda_n + 4| \approx 2 \Rightarrow \lambda_n = -2 \text{ o } -6$$

$$u_n \approx [1.0000, 1.0000, 0.0001]^T$$

$$(A + 2I)x^{(7)} = [0, 0, 0.0008]^T \Rightarrow \lambda_n = -2$$



Sistemas de Ecuaciones Lineales

Descomposición QR de una matriz.

Definición. Una matriz Q se dice que es ortogonal si cumple $Q Q^t = Q^t Q = I$

Obs. Para una matriz Q ortogonal se cumple $Q^{-1} = Q^t$

Teorema. Sea $A \in R^{n \times n}$ no singular, entonces ésta puede ser expresada como

$$A = QR$$

donde Q es ortogonal y R triangular superior.

En Matlab:
[Q,R] = qr(A)

Ejemplo. Encontrar Q ortogonal y R triangular superior para $A = \begin{pmatrix} 1 & 2 \\ 1 & 3 \end{pmatrix}$

$$Q = \begin{pmatrix} -0.7071 & -0.7071 \\ -0.7071 & 0.7071 \end{pmatrix} \quad R = \begin{pmatrix} -1.4142 & -3.5355 \\ 0 & 0.7071 \end{pmatrix}$$

$$Q * Q^T = Q^T * Q = I$$

R es triangular superior

Sistemas de Ecuaciones Lineales

Matrices similares.

Definición. Sean A y $B \in R^{n \times n}$ no singular, decimos que A y B son similares si existe un matriz M no singular tal que se cumple $A = M^{-1} B M$

Obs. Si A y B son matrices similares, entonces tienen el mismo polinomio característico. Sigue del hecho:

$$A - \lambda I = M^{-1} B M - \lambda I = M^{-1} (B - \lambda I) M$$

Ejemplo. Sean $A = \begin{pmatrix} 8 & 2 \\ 2 & 5 \end{pmatrix}$ y $B = \begin{pmatrix} 8.7647 & -1.0588 \\ -1.0588 & 4.2353 \end{pmatrix}$

A y B son similares, ya que existe M tal que $A = M^{-1} B M$

$$M = \begin{pmatrix} -0.9701 & -0.2425 \\ -0.2425 & 0.9701 \end{pmatrix}$$

en MATLAB:

```
A = sym('[8,2;2,5]')
```

```
pcA = poly(A)
```

```
B = sym('8.7647,-1.0588;-1.0588,4.2353')
```

```
pcB = poly(B)
```

$$M^{-1} B M = \begin{pmatrix} -0.9701 & -0.2425 \\ -0.2425 & 0.9701 \end{pmatrix}^{-1} \begin{pmatrix} 8.7647 & -1.0588 \\ -1.0588 & 4.2353 \end{pmatrix} \begin{pmatrix} -0.9701 & -0.2425 \\ -0.2425 & 0.9701 \end{pmatrix} = \begin{pmatrix} 8 & 2 \\ 2 & 5 \end{pmatrix}$$

Sistemas de Ecuaciones Lineales

Método QR para cálculo de autovalores.

Leer A , $itmax$

$$A_0 = A$$

Para $k = 1$ hasta $itmax$

met_qr.m

Calcular Q_k y R_k tal que $A_{k-1} = Q_k R_k$

Calcular $A_k = R_k Q_k$

Fin para

Obs. La matriz que genera el método, es decir A_k , converge bajo ciertas condiciones, a una matriz triangular superior, donde los autovalores aparecen sobre la diagonal principal.

Obs. $A_k = Q_k^T A_{k-1} Q_k$

$$A_k = R_k A_{k-1} R_k^{-1}$$

Los A_k todos tienen los mismos autovalores.

Todas son matrices similares.

Sistemas de Ecuaciones Lineales

Método QR para cálculo de autovalores.

Ejemplo. $A = \begin{pmatrix} 8 & 2 \\ 2 & 5 \end{pmatrix}$ con autovalores $\lambda_1 = 9, \lambda_2 = 4$

Descomponer $A_0 = A = Q_1 R_1$

$$Q_1 = \begin{pmatrix} -0.9701 & -0.2425 \\ -0.2425 & 0.9701 \end{pmatrix} \Rightarrow A_1 = R_1 Q_1 \cong \begin{pmatrix} 8.7647 & -1.0588 \\ -1.0588 & 4.2353 \end{pmatrix}$$

$$R_1 = \begin{pmatrix} 8.2462 & -3.1530 \\ 0 & 4.3656 \end{pmatrix}$$

Descomponer $A_1 = Q_2 R_2$

$$Q_2 = \begin{pmatrix} -0.9928 & 0.1199 \\ 0.1199 & 0.9928 \end{pmatrix} \Rightarrow A_2 = R_2 Q_2 \cong \begin{pmatrix} 8.9517 & 0.4891 \\ 0.4891 & 4.0483 \end{pmatrix}$$

$$R_2 = \begin{pmatrix} -8.8284 & 1.5591 \\ 0 & 4.0777 \end{pmatrix}$$

⋮

Descomponer $A_4 = Q_5 R_5$

$$Q_5 = \begin{pmatrix} -0.9999 & -0.108 \\ -0.0108 & 0.9999 \end{pmatrix} \Rightarrow A_5 = R_5 Q_5 \cong \begin{pmatrix} 8.996 & -0.0434 \\ -0.0434 & 4.0004 \end{pmatrix}$$

$$R_5 = \begin{pmatrix} -8.9986 & -0.1409 \\ 0 & 4.0006 \end{pmatrix}$$

Sistemas de Ecuaciones Lineales

Método QR para cálculo de autovalores.

Obs.

- En el algoritmo del método *QR* se puede utilizar como criterio de parada la verificación de que la matriz A_k es triangular superior.
- Si todos los autovalores de A tienen distinto módulo, es decir

$$|\lambda_1| > |\lambda_2| > \dots > |\lambda_{n-1}| > |\lambda_n| > 0,$$

el límite de la sucesión es una matriz triangular superior en la que los elementos de su diagonal son los autovalores de la matriz.

- Si existen autovalores de A de igual módulo la matriz límite es una matriz triangular superior por bloques, en la que cada bloque de orden k de su diagonal es una matriz cuyos autovalores son todos los k autovalores de igual módulo de la matriz A .

$$A = \begin{pmatrix} 2 & 3 & 5 \\ 1 & 4 & 6 \\ -1 & 3 & 1 \end{pmatrix} \quad \text{autovalores:} \quad 1.4142, -1.4142, 7.0000 \quad A_{1000} = \begin{pmatrix} 7 & -3.8378 & -5.5850 \\ 0 & 1.4456 & -0.0527 \\ 0 & 1.7021 & -1.4456 \end{pmatrix}$$

Matrices en MATLAB

Funciones que actúan sobre matrices: A una matriz de dimensión $m \times n$

- **triu**: extrae el triángulo superior de una matriz
triu(A) es el triangulo superior de la matriz A
triu(A,k) son los elementos en y encima de la diagonal k de la matriz A. Si $k = 0$ es respecto a la diagonal principal, si $k > 0$ es por encima de la diagonal principal, si $k < 0$ es por debajo de la diagonal principal.
- **tril** extrae el triángulo inferior de una matriz.
tril(A) es el triangulo inferior de la matriz A
tril(A,k) son los elementos en y debajo de la diagonal k de la matriz A. Si $k = 0$ es respecto a la diagonal principal, si $k > 0$ es por encima de la diagonal principal, si $k < 0$ es por debajo de la diagonal principal.

Ejemplo: Para $A = [1 \ -1 \ 2; \ 2 \ -3 \ 4; \ 5 \ 1 \ 2]$, calcular
triu(A,1), triu(A,-1), tril(A,1), tril(A,-1)

Sistemas de Ecuaciones Lineales

Método QR para cálculo de autovalores.

Ejemplo.

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

Al aplicarle el método QR después de 100 iteraciones se obtiene:

$$A_{100} = \begin{pmatrix} 22.7159 & 1.4858 & 3.1296 & -5.0027 & -4.4042 \\ 0 & -6.6466 & 4.6148 & 1.1541 & -2.4864 \\ 0 & 0 & 3.9861 & -2.8897 & 5.1189 \\ 0 & 0 & 0 & 0.1132 & -4.9856 \\ 0 & 0 & 0 & 0.7807 & 0.8314 \end{pmatrix}$$

Los autovalores de A:

22.7159

-6.6466

0.4723 + 1.9399i

0.4723 - 1.9399i

3.9861

} mismo módulo

Sistemas de Ecuaciones Lineales

Método QR para cálculo de autovalores.

Obs.

- El costo de aplicar la descomposición QR es “alto” (este requiere el doble del número de operaciones elementales del método de descomposición LU).
- Si $A = QR$, Q ortogonal y R triangular inferior, entonces

$$\det(A) = \pm \det(R) = \pm \prod_{i=1}^n r_{ii}$$

ya que $\det(Q) = \pm 1$.

- La convergencia del método QR puede ser lenta
- Dada la lentitud del método QR , se plantea como técnica para acelerarlo, la transformación de la matriz A a una matriz similar con estructura Hessenberg superior.

En Matlab: `>> A = hess(A);`

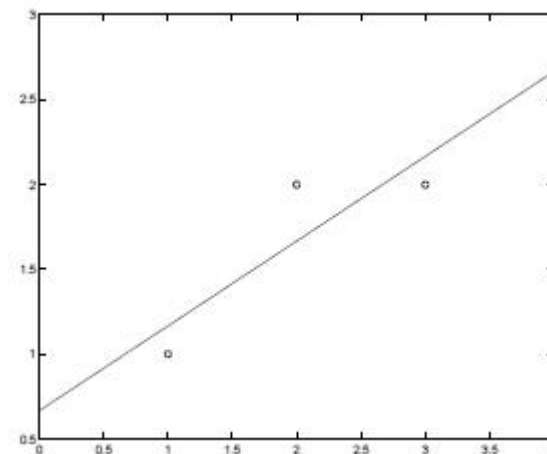
Luego, es a ésta matriz Hessenberg superior a la cual se le aplica el método QR .

Aproximación de Funciones

Ajustando una recta

Tabla de datos:

x	1	2	3
y	1	2	2



Sea $y^*(x) = c_0 + c_1x$. ¡Determine los parámetros c_0 , c_1 !

Sería ideal que:

$$y^*(1) = y(1) \quad \Rightarrow \quad c_0 + c_1 = 1$$

$$y^*(2) = y(2) \quad \Rightarrow \quad c_0 + 2c_1 = 2$$

$$y^*(3) = y(3) \quad \Rightarrow \quad c_0 + 3c_1 = 2$$

“Sistema sobredeterminado” — ¿Cuál es su solución?

Tres ecuaciones, dos incógnitas c_0 , c_1 .

Aproximación de Funciones

Ajustando una recta

Aproximación “minimax”

Error $e(x) = y^*(x) - y(x) = c_0 + c_1x - y(x)$.

¡Determine c_0, c_1 de modo de minimizar $\|e(x)\|_\infty$!

$$\min_{c_0, c_1} \|e\|_\infty \quad \Rightarrow \quad |e(1)| = |e(2)| = |e(3)|$$

$$e(1) = -e(2) \quad \Rightarrow \quad c_0 + c_1 - 1 = -(c_0 + 2c_1 - 2)$$

$$e(1) = e(3) \quad \Rightarrow \quad c_0 + c_1 - 1 = c_0 + 3c_1 - 2$$

$$2c_0 + 3c_1 = 3$$

$$2c_1 = 1 \quad \Rightarrow \quad c_1 = 1/2, \quad c_0 = 3/4$$

Mejor aproximación (minimax): $y^*(x) = \frac{3}{4} + \frac{1}{2}x$

Aproximación de Funciones

Ajustando una recta

Aproximación “mínimos cuadrados”

Determine c_0, c_1 **para minimizar** $\rho(c_0, c_1) = \|e(x)\|_2^2$

$$\rho = (c_0 + c_1 - 1)^2 + (c_0 + 2c_1 - 2)^2 + (c_0 + 3c_1 - 2)^2$$

$$\min_{c_0, c_1} \rho(c_0, c_1) = \min_{c_0, c_1} \sum_i |e(x_i)|_2^2$$

Mínimo “suave” si $\partial\rho/\partial c_0 = \partial\rho/\partial c_1 = 0$

$$\partial\rho/\partial c_0 := 0 \quad \Rightarrow \quad 6c_0 + 12c_1 - 10 = 0$$

$$\partial\rho/\partial c_1 := 0 \quad \Rightarrow \quad 12c_0 + 28c_1 - 22 = 0$$

Solución: $c_1 = 1/2, c_0 = 2/3$

Mejor aproximación (mínimos cuadrados):

$$y^*(x) = \frac{2}{3} + \frac{1}{2}x$$

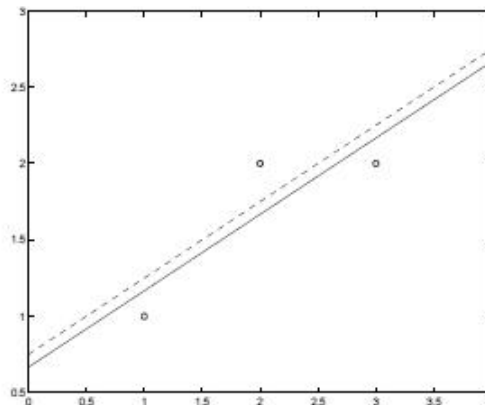
Aproximación de Funciones

Ajustando una recta

Línea punteada:

minimax

$$y^*(x) = \frac{3}{4} + \frac{1}{2}x$$



Datos				
x	1	2	3	
y	1	2	2	
Solución minimax				
y^*	5/4	7/4	9/4	
e	1/4	-1/4	1/4	$\ e\ _\infty = 1/4$ $\ e\ _2^2 = 3/16$
Solución mínimos cuadrados				
y^*	7/6	10/6	13/6	
e	1/6	-2/6	1/6	$\ e\ _\infty = 1/3$ $\ e\ _2^2 = 3/18$

Comparación

La solución depende de la escogencia de la norma

Línea sólida:
mínimos cuadrados

$$y^*(x) = \frac{2}{3} + \frac{1}{2}x$$

“minimax” y
“mínimos cuadrados”
no son lo mismo

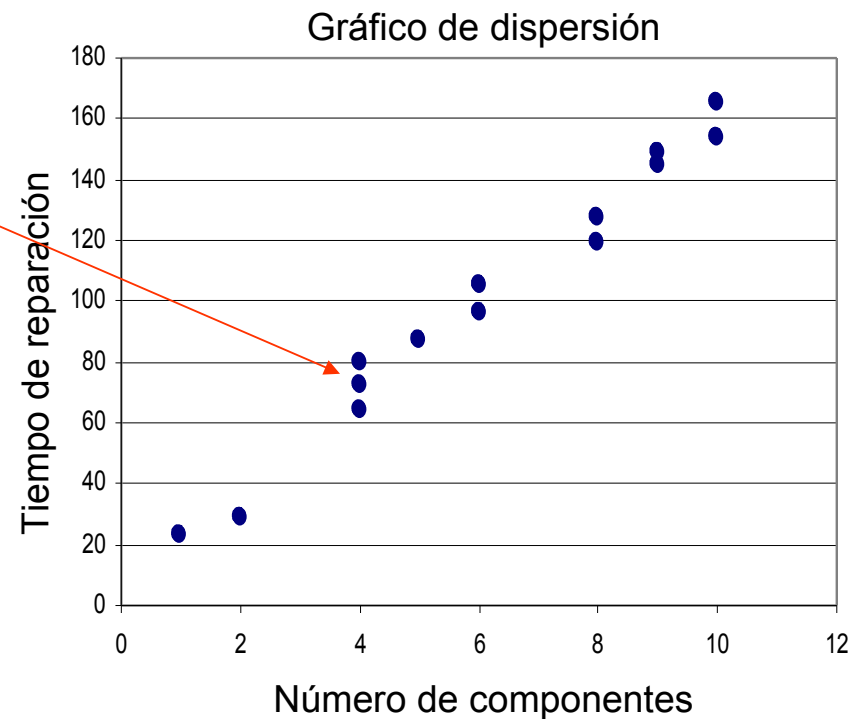


Aproximación de Funciones

Ejemplo

Una compañía que repara computadoras pequeñas requiere desarrollar un mecanismo para proveer a sus clientes del tiempo estimado de reparación.

observación	Número de componentes	Tiempo de reparación
1	1	23
2	2	29
3	4	64
4	4	72
5	4	80
6	5	87
7	6	96
8	6	105
9	8	127
10	8	119
11	9	145
12	9	149
13	10	165
14	10	154



Aproximación de Funciones

Método de mínimos cuadrados

Consideremos el problema de estimar los valores de una función en puntos no tabulados, dados los datos experimentales, como en la tabla anterior.

La gráfica de los valores dados nos indica que sería razonable suponer que la relación es lineal y que ninguna recta se ajusta a los datos exactamente debido al error en el procedimiento de recolección de datos.

El mejor enfoque sería encontrar la “mejor” recta que se pudiera usar como función aproximante, aún cuando pudiera no coincidir precisamente con los datos en cada punto.

El enfoque de mínimos cuadrados a este problema requiere de la determinación de la mejor recta aproximante cuando el error involucrado es la suma de los cuadrados de las diferencias entre los valores de la recta aproximante y los valores dados.

Aproximación de Funciones

Método de mínimos cuadrados

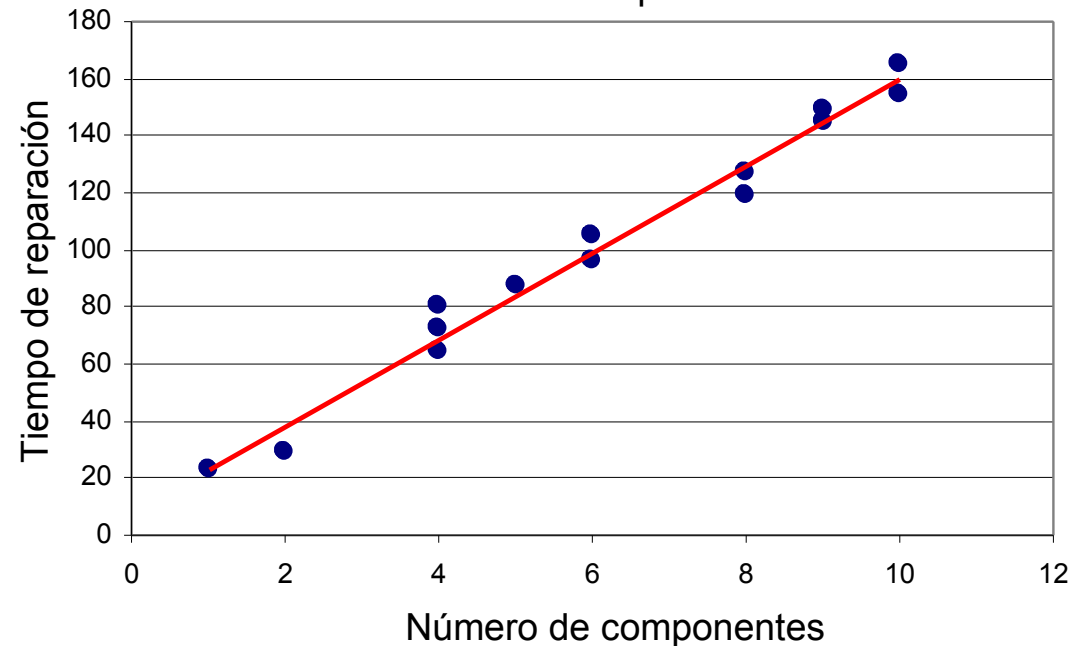
Ejemplo (cont.)

observación Número de componentes Tiempo de reparación

i	X_i	Y_i
1	1	23
2	2	29
3	4	64
4	4	72
5	4	80
6	5	87
7	6	96
8	6	105
9	8	127
10	8	119
11	9	145
12	9	149
13	10	165
14	10	154

$$y = 15.198x + 7.711$$

Gráfico de dispersión

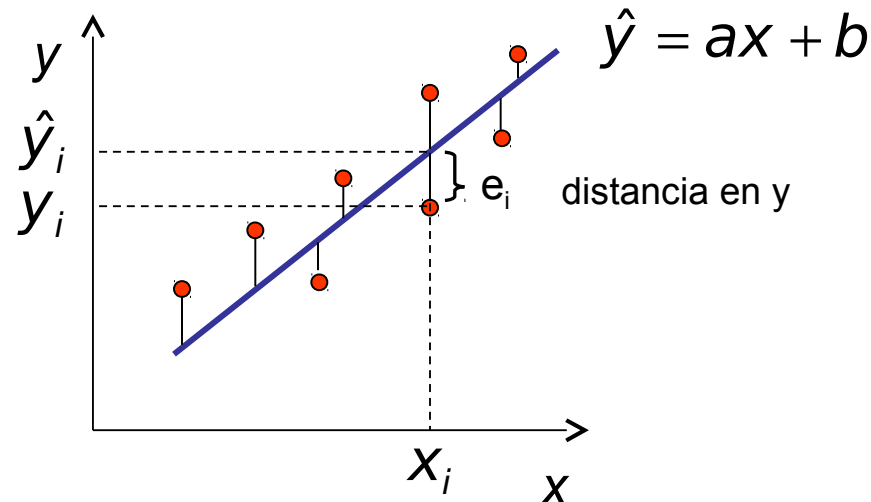


Aproximación de Funciones

Método de mínimos cuadrados

Denotando por $\hat{y}_i = ax_i + b$ al i -ésimo valor de la recta aproximante y por y_i al i -ésimo valor dado. Se necesita encontrar las constantes a y b tales que minimizan el error E de mínimos cuadrados:

$$E = \sum_{i=1}^n e_i^2 = \sum_{i=1}^n (y_i - \hat{y}_i)^2 = \sum_{i=1}^n (y_i - ax_i - b)^2$$



Aproximación de Funciones

Método de mínimos cuadrados

Si consideramos a E como una función de las 2 variables a y b ,

$$E(a, b) = \sum_{i=1}^n (y_i - ax_i - b)^2$$

un resultado elemental del cálculo de varias variables implica, para que un mínimo ocurra en (a, b) , es necesario que

$$0 = \frac{\partial E(a, b)}{\partial a} = \frac{\partial}{\partial a} \sum_{i=1}^n (y_i - ax_i - b)^2 \quad \text{y} \quad 0 = \frac{\partial E(a, b)}{\partial b} = \frac{\partial}{\partial b} \sum_{i=1}^n (y_i - ax_i - b)^2$$

$$\Rightarrow \quad 0 = 2 \sum_{i=1}^n (y_i - ax_i - b)(-x_i) \quad \text{y} \quad 0 = 2 \sum_{i=1}^n (y_i - ax_i - b)(-1).$$

Estas ecuaciones se simplifican a lo que se conoce como **ecuaciones normales**

$$a \sum_{i=1}^n x_i^2 + b \sum_{i=1}^n x_i = \sum_{i=1}^n x_i y_i \quad \text{y} \quad a \sum_{i=1}^n x_i + bn = \sum_{i=1}^n y_i$$

Aproximación de Funciones

Método de mínimos cuadrados

La solución de este sistema de ecuaciones es

$$a = \frac{n \left(\sum_{i=1}^n x_i y_i \right) - \left(\sum_{i=1}^n x_i \right) \left(\sum_{i=1}^n y_i \right)}{n \left(\sum_{i=1}^n x_i^2 \right) - \left(\sum_{i=1}^n x_i \right)^2} \quad y \quad b = \frac{1}{n} \sum_{i=1}^n y_i - a \frac{1}{n} \sum_{i=1}^n x_i$$

Usando las definiciones de promedio, varianza y covarianza siguientes

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad S_{xx} = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2 \quad S_{yx} = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})$$

promedio varianza covarianza

se tiene que las variables a y b se pueden calcular como

$$a = \frac{S_{yx}}{S_{xx}} \quad y \quad b = \bar{y} - a\bar{x}$$

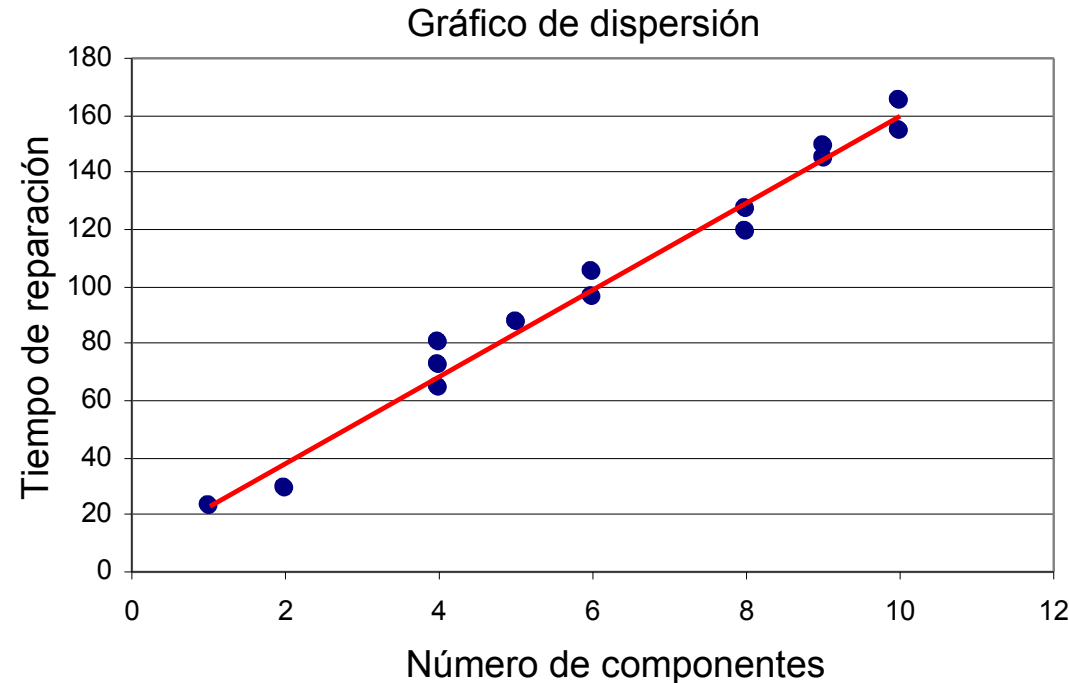
donde $x = (x_1, \dots, x_n)$ $y = (y_1, \dots, y_n)$

Aproximación de Funciones

Método de mínimos cuadrados

Ejemplo (cont.)

observación	Número de componentes	Tiempo de reparación
i	X_i	y_i
1	1	23
2	2	29
3	4	64
4	4	72
5	4	80
6	5	87
7	6	96
8	6	105
9	8	127
10	8	119
11	9	145
12	9	149
13	10	165
14	10	154
suma	86	1415
promedio	6,14	101,07



$$S_{yx} = 130.60 \quad S_{xx} = 7.98$$

$$a = \frac{S_{yx}}{S_{xx}} = 15.198 \quad b = \bar{y} - a\bar{x} = 7.711$$

$$y = 15.198x + 7.711$$

Aproximación de Funciones

Método de mínimos cuadrados

El problema general de aproximar un conjunto de datos

$$\{ (x_i, y_i) : i = 0, \dots, m \}$$

con un polinomio de grado $n < m$

$$P_n(x) = \sum_{i=0}^n a_i x^i$$

usando el procedimiento de mínimos cuadrados se trata de manera similar y requiere de la elección de las constantes a_0, a_1, \dots, a_n para minimizar el error de mínimos cuadrados

$$\begin{aligned} E &= \sum_{i=0}^m (y_i - P(x_i))^2 = \sum_{i=0}^m y_i^2 - 2 \sum_{i=0}^m P(x_i) y_i + \sum_{i=0}^m (P(x_i))^2 \\ &= \sum_{i=0}^m y_i^2 - 2 \sum_{i=0}^m \left(\sum_{j=0}^n a_j x_i^j \right) y_i + \sum_{i=0}^m \left(\sum_{j=0}^n a_j x_i^j \right)^2 \\ &= \sum_{i=0}^m y_i^2 - 2 \sum_{j=0}^n a_j \left(\sum_{i=0}^m y_i x_i^j \right) + \sum_{j=0}^n \sum_{k=0}^n a_j a_k \left(\sum_{i=0}^m x_i^{j+k} \right) \end{aligned}$$

Aproximación de Funciones

Método de mínimos cuadrados

Como en el caso lineal para minimizar E como función de los a_j , es necesario que $\partial E / \partial a_j = 0$ para cada $j = 0, \dots, n$.

Por lo tanto para cada j

$$0 = \frac{\partial E}{\partial a_j} = -2 \sum_{i=0}^m y_i x_i^j + 2 \sum_{k=0}^n a_k \left(\sum_{i=0}^m x_i^{j+k} \right)$$

Esto da $n+1$ ecuaciones para las $n+1$ incógnitas a_j , que se conocen como **ecuaciones normales**,

$$\sum_{k=0}^n a_k \left(\sum_{i=0}^m x_i^{j+k} \right) = \sum_{i=0}^m y_i x_i^j \quad \text{para } j = 0, \dots, n.$$

Aproximación de Funciones

Método de mínimos cuadrados

Estas ecuaciones se pueden escribir como

$$a_0 \sum_{i=0}^m x_i^0 + a_1 \sum_{i=0}^m x_i^1 + \cdots + a_n \sum_{i=0}^m x_i^n = \sum_{i=0}^m y_i x_i^0$$

$$a_0 \sum_{i=0}^m x_i^1 + a_1 \sum_{i=0}^m x_i^2 + \cdots + a_n \sum_{i=0}^m x_i^{n+1} = \sum_{i=0}^m y_i x_i^1$$

...

$$a_0 \sum_{i=0}^m x_i^n + a_1 \sum_{i=0}^m x_i^{n+1} + \cdots + a_n \sum_{i=0}^m x_i^{2n} = \sum_{i=0}^m y_i x_i^n$$

y escrito en forma matricial

$$\begin{pmatrix} \sum_{i=0}^m x_i^0 & \cdots & \sum_{i=0}^m x_i^n \\ \vdots & \ddots & \vdots \\ \sum_{i=0}^m x_i^n & \cdots & \sum_{i=0}^m x_i^{2n} \end{pmatrix} \begin{pmatrix} a_0 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} \sum_{i=0}^m y_i x_i^0 \\ \vdots \\ \sum_{i=0}^m y_i x_i^n \end{pmatrix}$$

Se puede demostrar que este sistema tiene solución única siempre que las x_i , para $i=0, 1, \dots, m$ sean distintas.

Aproximación de Funciones

Método de mínimos cuadrados

Ejemplo:

i	0	1	2	3	4
X_i	0	0.25	0.5	0.75	1.00
Y_i	1.00	1.248	1.6487	2.117	2.7183

Ajustar con un polinomio de mínimos cuadrados de grado 2

$$P(x) = a_0x^0 + a_1x^1 + a_2x^2$$

$$\begin{pmatrix} \sum_{i=0}^4 x_i^0 & \sum_{i=0}^4 x_i^1 & \sum_{i=0}^4 x_i^2 \\ \sum_{i=0}^4 x_i^1 & \sum_{i=0}^4 x_i^2 & \sum_{i=0}^4 x_i^3 \\ \sum_{i=0}^4 x_i^2 & \sum_{i=0}^4 x_i^3 & \sum_{i=0}^4 x_i^4 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} \sum_{i=0}^4 y_i x_i^0 \\ \sum_{i=0}^4 y_i x_i^1 \\ \sum_{i=0}^4 y_i x_i^2 \end{pmatrix}$$

Las ecuaciones normales

$$\begin{pmatrix} 5 & 2.5 & 1.87 \\ 2.5 & 1.875 & 1.5625 \\ 1.875 & 1.5625 & 1.3828 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} 8.732 \\ 5.4424 \\ 4.3993 \end{pmatrix}$$

Aproximación de Funciones

Método de mínimos cuadrados

Ejemplo (cont.):

Las ecuaciones normales:

$$5a_0 + 2.5a_1 + 1.875a_2 = 8.732$$

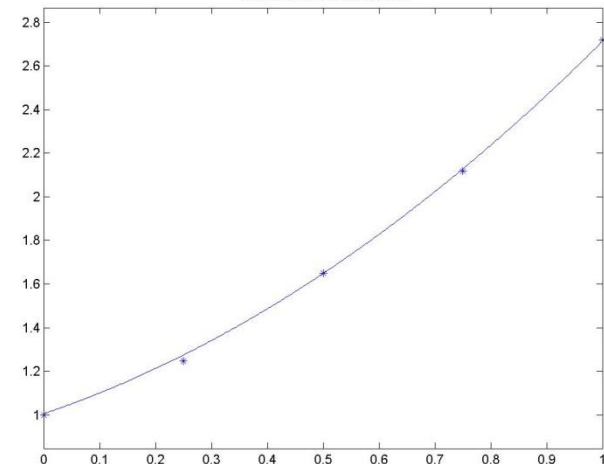
$$2.5a_0 + 1.875a_1 + 1.5625a_2 = 5.4424$$

$$1.875a_0 + 1.5625a_1 + 1.3828a_2 = 4.3992875$$

La solución del sistema $a_0 = 0.9959$, $a_1 = 0.8374$, $a_2 = 0.8848$

$P(x_i)$	0.9959	1.2606	1.6358	2.1217	2.7181
----------	--------	--------	--------	--------	--------

El error $E = \sum_{i=0}^m (y_i - P(x_i))^2 = 3.624 \times 10^{-4}$



Aproximación de Funciones

Método de mínimos cuadrados

Ejemplo:

$$x = 0:0.1:1;$$

$$y = [-0.447, 1.978, 3.28, 6.16, 7.08, 7.34, 7.66, 9.56, 9.48, 9.30, 11.2];$$

Ajustar polinomios de grado 2 y 8 para los datos.

Representar sobre una misma gráfica los puntos y ambos polinomios.

¿Qué se observa?

Aproximación de Funciones

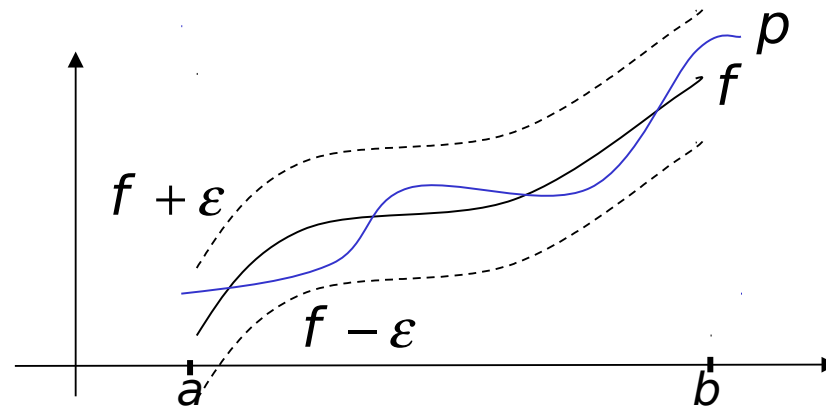
La clase de los polinomios reales es el conjunto de funciones de la forma

$$p(x) = a_0 + a_1x + a_2x^2 + \cdots + a_nx^n.$$

Teorema de aproximación de Weierstrass

Si f es una función $[a,b] \subset \mathbb{R} \rightarrow \mathbb{R}$ y continua en $[a,b]$, dado $\varepsilon > 0$, existe un polinomio p , definido en $[a,b]$, con la propiedad de que

$$|f(x) - p(x)| < \varepsilon \quad \text{para todo } x \in [a,b].$$



Aspectos importantes para considerar la clase de polinomios en la aproximación de funciones:

- es sencillo determinar la derivada
- es sencillo determinar la integral indefinida

En ambos casos el resultado sigue siendo un polinomio.

Aproximación de Funciones

Polinomio de Taylor

Consideremos el problema de encontrar un polinomio de grado específico que este “cerca” de una función dada, alrededor de un punto.

Un polinomio p coincidirá con una función f en el punto x_0 cuando

$$p(x_0) = f(x_0).$$

El polinomio tendrá la misma “dirección” que la función f en $(x_0, f(x_0))$ si

$$p'(x_0) = f'(x_0).$$

El polinomio de grado n que mejor aproxime a la función f cerca de x_0 , tendrá tantas derivadas en x_0 , como sea posible que coincidan con las de f .

Esta es precisamente la condición que satisface el polinomio de Taylor de grado n para la función f en x_0 :

$$p_n(x) = f(x_0) + f'(x_0)(x - x_0) + f''(x_0)\frac{(x - x_0)^2}{2!} + \dots + f^{(n)}(x_0)\frac{(x - x_0)^n}{n!},$$

el cual tiene un término de error $p_n(x) - f(x) = R_n(x) = f^{(n+1)}(\xi)\frac{(x - x_0)^{n+1}}{(n+1)!}$,

para algún número $\xi(x)$ entre x_0 y x .

Aproximación de Funciones

Polinomio de Taylor

Ejemplo:

El polinomio de Taylor de grado 3 alrededor de $x_0=0$ para $f(x)=(1+x)^{1/2}$

$$p_3(x) = f(0) + f'(0)x + f''(0)\frac{x^2}{2!} + f'''(0)\frac{x^3}{3!} = 1 + \frac{1}{2}x - \frac{1}{8}x^2 + \frac{1}{16}x^3$$

Si queremos aproximar

$$\sqrt{1.1} = f(0.1) \approx p_3(0.1) = 1 + \frac{1}{2}0.1 - \frac{1}{8}(0.1)^2 + \frac{1}{16}(0.1)^3 = 1.0488125$$

donde el error

$$\begin{aligned} |R_3(0.1)| &= \left| f^{(4)}(\xi) \frac{(0.1)^4}{4!} \right| = \left| -\frac{15}{16} (1+\xi)^{-7/2} \right| \frac{(0.1)^4}{24} \\ &\leq \frac{15}{16 \cdot 24} (0.1)^4 \max_{\xi \in [0,0.1]} (1+\xi)^{-7/2} \leq 3.91 \times 10^{-6} \end{aligned}$$

Aproximación de Funciones

Polinomio de Taylor

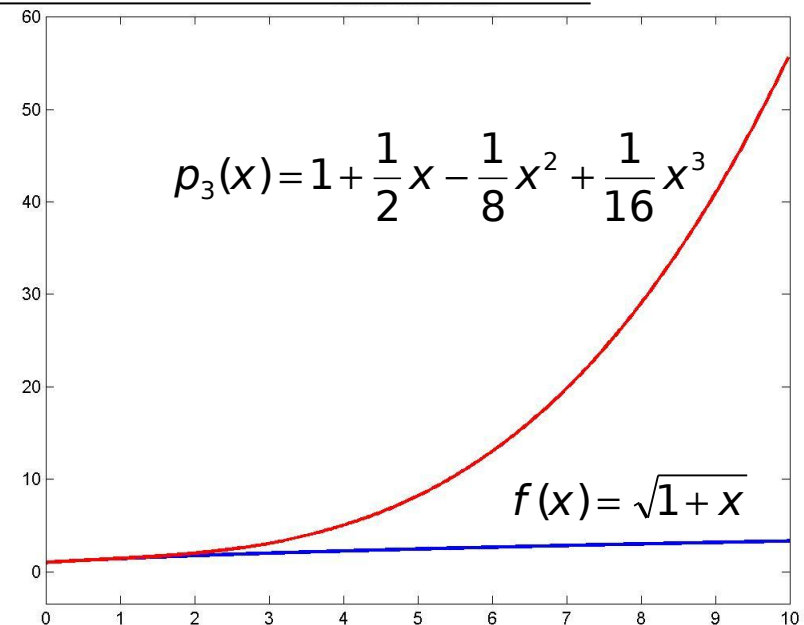
Ejemplo (cont.):

x	0.1	0.5	1	2	10
$p_3(x)$	1.048813	1.2266	1.438	2.00	56.00
$f(x)$	1.048809	1.2247	1.414	1.73	3.32
$ p_3(x) - f(x) $	0.000004	0.0019	0.024	0.27	52.68

$$R_n(x) = f^{(n+1)}(\xi) \frac{(x - x_0)^{n+1}}{(n+1)!},$$

La razón para que esta técnica de aproximación falle es que el término del error crece en valor absoluto conforme n crece.

Esto resulta de que $x=10$ no está lo suficientemente cerca de $x_0=0$.



Aproximación de Funciones

Función “taylor”

Calcula el desarrollo en serie de Taylor alrededor de un punto dado, para una función simbólica dada. Por defecto el polinomio es de grado 5 y alrededor del cero.

Polinomio de Taylor de grado n alrededor del punto x_0 :

$$p_n(x) = f(x_0) + f'(x_0)(x - x_0) + f''(x_0)\frac{(x - x_0)^2}{2!} + \dots + f^{(n)}(x_0)\frac{(x - x_0)^n}{n!},$$

el cual tiene un término de error $p_n(x) - f(x) = R_n(x) = f^{(n+1)}(\xi)\frac{(x - x_0)^{n+1}}{(n+1)!}$,

para algún número $\xi(x)$ entre x_0 y x .

El polinomio de Taylor de grado 3 alrededor de $x_0=0$ para $f(x)=(1+x)^{1/2}$

$$p_3(x) = f(0) + f'(0)x + f''(0)\frac{x^2}{2!} + f'''(0)\frac{x^3}{3!} = 1 + \frac{1}{2}x - \frac{1}{8}x^2 + \frac{1}{16}x^3$$

>> `taylor(sqrt(1+x), 8, 0)` →

$$1 + 1/2*x - 1/8*x^2 + 1/16*x^3 - 5/128*x^4 + 7/256*x^5 - 21/1024*x^6 + 33/2048*x^7$$

Aproximación de Funciones

Función “taylor” (cont.)

```
>> syms x;
```

```
>> taylor(exp(-x), 6, 0) retorna
```

$$\text{ans} = 1 - x + \frac{1}{2}x^2 - \frac{1}{6}x^3 + \frac{1}{24}x^4 - \frac{1}{120}x^5$$

```
>> taylor(log(x), 6, 1) retorna (grado 5 y alrededor del 1)
```

$$\text{ans} = x - 1 - \frac{1}{2}(x-1)^2 + \frac{1}{3}(x-1)^3 - \frac{1}{4}(x-1)^4 + \frac{1}{5}(x-1)^5$$

```
>> taylor(sin(x), 8, pi/2) retorna (grado 7 y alrededor de pi/2)
```

$$\text{ans} = 1 - \frac{1}{2}(x - \frac{1}{2}\pi)^2 + \frac{1}{24}(x - \frac{1}{2}\pi)^4 - \frac{1}{720}(x - \frac{1}{2}\pi)^6$$

```
>> syms t;
```

```
>> taylor(x^t, 3, t) retorna
```

(grado 2, respecto a la variable t y alrededor de 0)

$$\text{ans} = 1 + \log(x)t + \frac{1}{2}\log(x)^2 t^2$$

Aproximación de Funciones

Polinomio de Taylor

Ejemplo: Aproximar $\ln 2$ con un error absoluto menor que 10^{-8} .

Sea $f(x) = \ln x$ definida en $[1,2]$. Así, $f(x) \in C^\infty[1,2]$.

Por el teorema de Taylor, se tiene:

$$f(x) = p_n(x) + R_n(x), \text{ para cualquier } x \in [1,2] \text{ y cualquier } n \in \mathbb{N},$$

donde

$$p_n(x) = \sum_{k=0}^n f^{(k)}(x_0) \frac{(x - x_0)^k}{k!} \quad \text{y} \quad R_n(x) = f^{(n+1)}(\xi) \frac{(x - x_0)^{n+1}}{(n+1)!},$$

con $x_0 = 1$ y $1 < \xi < x$ (polinomio de Taylor alrededor de $x_0 = 1$)

Se tiene $f^{(k)}(x) = (-1)^{k-1} (k-1)! x^{-k}$, $k \geq 1 \Rightarrow f^{(k)}(1) = (-1)^{k-1} (k-1)!$

de donde

$$p_n(x) = \sum_{k=1}^n \frac{(-1)^{k-1} (k-1)!}{k!} (x-1)^k = \sum_{k=1}^n \frac{(-1)^{k-1}}{k} (x-1)^k$$

$$|R_n(x)| = \left| \frac{(-1)^n n! \xi^{-(n+1)}}{(n+1)!} (x-1)^{n+1} \right| = \frac{1}{n+1} \left(\frac{1}{\xi} \right)^{n+1} (x-1)^{n+1}$$

Aproximación de Funciones

Polinomio de Taylor

Ejemplo (cont.): Aproximar $\ln 2$ con un error absoluto menor que 10^{-8} .

$$f(2) = p_n(2) + R_n(2) \Rightarrow \ln 2 = f(2) \approx p_n(2) \quad \text{para } n \text{ grande}$$

$$|R_n(2)| = \frac{1}{n+1} \left(\frac{1}{\xi} \right)^{n+1} (2-1)^{n+1} = \frac{1}{n+1} \left(\frac{1}{\xi} \right)^{n+1} \quad \text{con } 1 < \xi$$

$$\text{sigue} \quad \frac{1}{\xi} < 1 \Rightarrow |R_n(2)| < \frac{1}{n+1}$$

$$\text{Si se quiere } |R_n(2)| < 10^{-8}, \text{ basta que } \frac{1}{n+1} < 10^{-8} \Rightarrow n > 10^8 - 1$$

(100 millones de términos del polinomio de Taylor)

Usando el polinomio de Taylor alrededor de $x_0 = 3/2$, se tiene

$$|R_n(2)| = \left| \frac{(-1)^n n! \xi^{-(n+1)}}{(n+1)!} \left(2 - \frac{3}{2}\right)^{n+1} \right| = \frac{1}{n+1} \left(\frac{1}{\xi} \right)^{n+1} \left(\frac{1}{2} \right)^{n+1} < \frac{1}{n+1} \frac{1}{2^{n+1}}$$

$$\text{Si se quiere } |R_n(2)| < 10^{-8}, \text{ basta que } \frac{1}{n+1} \frac{1}{2^{n+1}} < 10^{-8} \Rightarrow n \approx 22$$

(aprox. 11 términos del polinomio de Taylor)

Aproximación de Funciones

Interpolación polinómica

Hemos usado polinomios aproximantes que coincidan con una función dada y con algunas de sus derivadas en un único punto.

Estos polinomios son útiles sobre intervalos pequeños para funciones cuyas derivadas existen y son fáciles de calcular, pero obviamente éste no es siempre el caso.

De aquí, el polinomio de Taylor es frecuentemente de poca utilidad, y se deben buscar métodos alternativos de aproximación.



Aproximación de Funciones

Formulación general del problema de interpolación polinómica:

Dados los puntos del plano $(x_0, f_0), (x_1, f_1), \dots, (x_n, f_n)$, donde los x_i son distintos. Determinar un polinomio p que satisfice

$$\begin{aligned} a) \text{ grado}(p) &\leq n \\ b) p(x_i) &= f_i, \quad i = 0, \dots, n. \end{aligned} \tag{7}$$

Escribiendo p en la base natural $1, t, \dots, t^n$ (polinomio interpolante natural)

$$p(t) = a_0 + a_1 t + a_2 t^2 + \dots + a_n t^n, \tag{8}$$

se obtiene el sistema lineal

$$\begin{pmatrix} 1 & x_0 & x_0^2 & \dots & x_0^n \\ 1 & x_1 & x_1^2 & \dots & x_1^n \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^n \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} f_0 \\ f_1 \\ \vdots \\ f_n \end{pmatrix} \tag{9}$$

Matriz de Vandermonde

Eliminación Gaussiana no se recomienda para resolver este sistema

Aproximación de Funciones

Interpolación polinómica

Ejemplo. En la determinación del polinomio cuadrático de interpolación para

$$x_0 = 100, \quad x_1 = 101, \quad x_2 = 102,$$

la matriz del sistema es

$$V = \begin{pmatrix} 1 & 100 & 10000 \\ 1 & 101 & 10201 \\ 1 & 102 & 10404 \end{pmatrix}$$

$$\begin{pmatrix} 1 & x_0 & x_0^2 \\ 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} f_0 \\ f_1 \\ f_2 \end{pmatrix}$$

$$V = [1, 100, 10000; 1, 101, 10201; 1, 102, 10404]$$

El número de condición de V es aproximadamente 2.20832×10^8 .

La presencia de la diferencia en escala de las entradas de la matriz sugiere el uso de técnicas de escalamiento para solventar el problema.

Escalando la matriz a la forma

$$\tilde{V} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1.01 & 1.0201 \\ 1 & 1.02 & 1.0404 \end{pmatrix}$$

$$V = [1, 1, 1; 1, 1.01, 1.0201; 1, 1.02, 1.0404]$$

conduce a un número de condición de 9.182×10^4 , resultando todavía grande.

Aproximación de Funciones

Formulación general del problema de interpolación polinómica:

$$x = [1, 2, 3, 4]; \quad y = [2, 3, 5, 8]$$

$$V = [1, 1, 1, 1; 1, 2, 4, 8; 1, 3, 9, 27; 1, 4, 16, 64]$$

$$V = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 \\ 1 & 3 & 9 & 27 \\ 1 & 4 & 16 & 64 \end{pmatrix}$$

El número de condición de V es aproximadamente 1.171×10^3 .

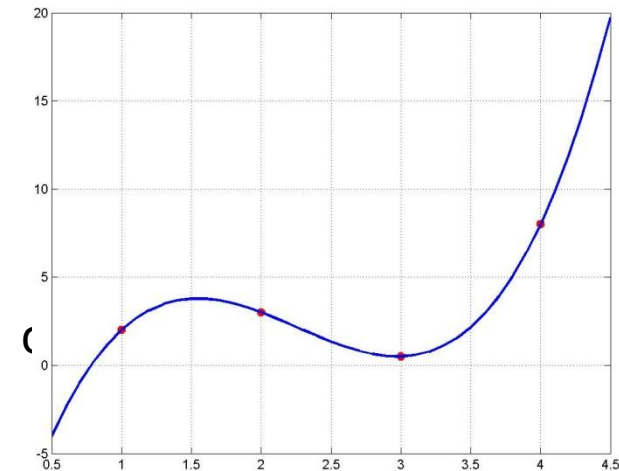
$$\text{sol} = \text{inv}(V) * y'$$

$$\rightarrow \text{sol} = -16.0000 \quad 31.0000 \quad -15.2500 \quad 2.2500$$

$$p(x) = 2.25x^3 - 15.25x^2 + 31x - 16$$

$$p = \text{polyfit}(x, y, 3)$$

$$\rightarrow p = 2.2500 \quad -15.2500 \quad 31.0000 \quad -16.0000$$



Aproximación de Funciones

Polinomio interpolante de Lagrange

Consideremos el problema de determinar un polinomio de grado 1 que pase por los puntos (x_0, y_0) y (x_1, y_1) . Este problema es el mismo que el de aproximar una función f , para la cual $f(x_0) = y_0$ y $f(x_1) = y_1$, por medio de un polinomio de grado 1, coincidiendo con los valores de f en los puntos dados.

La ecuación de la recta (polinomio de grado 1) que pasa por estos puntos

$$p(x) - y_0 = \frac{(y_1 - y_0)}{(x_1 - x_0)}(x - x_0) \Leftrightarrow p(x) = \frac{(x - x_1)}{(x_0 - x_1)}y_0 + \frac{(x - x_0)}{(x_1 - x_0)}y_1$$

donde en los casos $x = x_0$ y $x = x_1$ se obtiene $p(x_0) = f(x_0)$ y $p(x_1) = f(x_1)$, así p tiene las propiedades requeridas.

Definiendo $L_0(x) = \frac{(x - x_1)}{(x_0 - x_1)}$ y $L_1(x) = \frac{(x - x_0)}{(x_1 - x_0)}$ se tiene

$$p(x) = f(x_0)L_0(x) + f(x_1)L_1(x). \quad (10)$$

Aproximación de Funciones

Polinomio interpolante de Lagrange

Los polinomios lineales $L_0(x)$ y $L_1(x)$ satisfacen la propiedad

$$L_i(x_j) = \begin{cases} 1 & \text{si } i = j \\ 0 & \text{si } i \neq j \end{cases} \quad \text{para } 0 \leq i, j \leq 1 \quad (11)$$

Para el caso general, la existencia de un polinomio interpolante p puede ser establecida de la siguiente manera:

Supongamos que se tienen n polinomios $L_i(x)$ que satisfacen la condición

$$L_i(x_j) = \begin{cases} 1 & \text{si } i = j \\ 0 & \text{si } i \neq j \end{cases} \quad \text{para } 0 \leq i, j \leq n, \quad (12)$$

entonces el polinomio interpolante p tiene la forma

$$p(x) = f(x_0)L_0(x) + f(x_1)L_1(x) + \cdots + f(x_n)L_n(x) = \sum_{i=0}^n f(x_i)L_i(x). \quad (13)$$

Notamos que p satisface los requerimiento en (7), es decir, el grado es menor o igual a n y $p(x_i) = f(x_i)$.

Aproximación de Funciones

Polinomio interpolante de Lagrange

Los polinomios $L_j(x)$ (polinomio de Lagrange de grado n) se escogen de manera similar que en el caso lineal, para que satisfaga la condición (12)

$$L_j(x) = \frac{(x - x_0) \cdots (x - x_{j-1})(x - x_{j+1}) \cdots (x - x_n)}{(x_j - x_0) \cdots (x_j - x_{j-1})(x_j - x_{j+1}) \cdots (x_j - x_n)} \quad \text{para } 0 \leq j \leq n,$$

$$\Leftrightarrow L_j(x) = \prod_{\substack{i=0 \\ i \neq j}}^{i=n} \frac{x - x_i}{x_j - x_i} \quad \text{para } 0 \leq j \leq n. \quad (14)$$

Teorema. Si x_0, x_1, \dots, x_n son $(n+1)$ puntos diferentes y f es una función cuyos valores están dados en estos puntos, entonces existe un único polinomio p de grado a lo más n con la propiedad de que

$$f(x_k) = p(x_k) \quad \text{para } 0 \leq k \leq n. \quad (15)$$

El polinomio p está dado por (13), donde los $L_j(x)$ se definen en (14).

Obs. Una consecuencia de este resultado es que la matriz de Vandermonde (9) para $n+1$ puntos distintos es no singular.

Aproximación de Funciones

Polinomio interpolante de Lagrange

Obs. La unicidad del polinomio interpolante p está basado en el resultado siguiente:

Si un polinomio de grado n se anula en $n+1$ puntos distintos, entonces el polinomio es el polinomio nulo.

Supongamos que para el problema de interpolación existe otro polinomio q diferente al polinomio p . Entonces definimos el polinomio r de grado menor o igual a n como

$$r(x) = p(x) - q(x).$$

Como los polinomios p y q verifican la condición (15) se tiene

$$r(x_i) = p(x_i) - q(x_i) = f(x_i) - f(x_i) = 0,$$

de donde el polinomio r se anula en $n+1$ puntos. Entonces r es el polinomio nulo, y $p = q$.

Aproximación de Funciones

Polinomio interpolante de Lagrange

Teorema (término residual o cota de error). Si x_0, x_1, \dots, x_n son $(n+1)$ puntos diferentes en el intervalo $[a, b]$ y $f \in C^{n+1}[a, b]$, entonces para cada $x \in [a, b]$ existe un número $\xi(x)$ en (a, b) tal que

$$f(x) - p(x) = \frac{f^{(n+1)}(\xi(x))}{(n+1)!} (x - x_0) \cdots (x - x_n), \quad (16)$$

donde p es el polinomio interpolante dado por (13).

Obs. La forma del error del polinomio de Lagrange es similar a la del polinomio de Taylor.

El polinomio de Taylor de grado n alrededor de x_0 concentra toda la información conocida en x_0 , en cambio el polinomio de Lagrange de grado n usa información en los diferentes números x_0, x_1, \dots, x_n .

La fórmula del error (16) no es de uso práctico, ya que está restringida a funciones cuyas derivadas tengan cotas conocidas (por ejemplo funciones trigonométricas).

Aproximación de Funciones

Polinomio interpolante de Lagrange

Ejemplo. Para los puntos $x_0 = 2$, $x_1 = 2.5$ y $x_2 = 4$, determinar el polinomio interpolante de Lagrange de grado 2 para la función $f(x) = 1/x$.

Determinamos los polinomios L_0 , L_1 y L_2

$$L_0(x) = \frac{(x - 2.5)(x - 4)}{(2 - 2.5)(2 - 4)} = x^2 - 6.5x + 10$$

$$L_1(x) = \frac{(x - 2)(x - 4)}{(2.5 - 2)(2.5 - 4)} = \frac{1}{3}(-4x^2 + 24x - 32)$$

$$L_2(x) = \frac{(x - 2)(x - 2.5)}{(4 - 2)(4 - 2.5)} = \frac{1}{3}(x^2 - 4.5x + 5)$$

Como $f(x_0) = f(2) = 0.5$, $f(x_1) = f(2.5) = 0.4$, $f(x_2) = f(4) = 0.25$

$$\begin{aligned} p(x) &= \sum_{i=0}^2 f(x_i) L_i(x) \\ &= 0.5(x^2 - 6.5x + 10) + \frac{0.4}{3}(-4x^2 + 24x - 32) + \frac{0.25}{3}(x^2 - 4.5x + 5) \\ &= 0.05x^2 - 0.425x + 1.15 \end{aligned}$$

Aproximación de Funciones

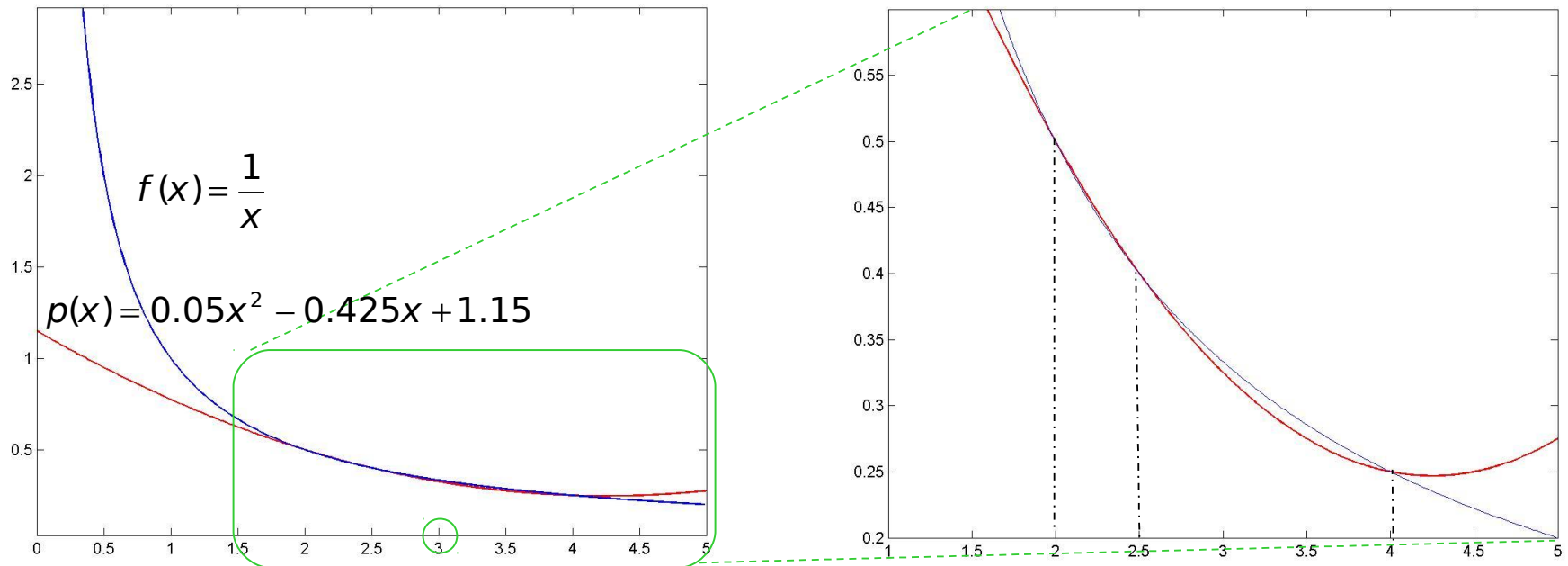
Polinomio interpolante de Lagrange

Ejemplo (cont.).

$$p(x) = 0.05x^2 - 0.425x + 1.15$$

$$f(x) = \frac{1}{x}$$

Una aproximación a $f(3) = \frac{1}{3}$ es $p(3) = 0.05 \cdot 3^2 - 0.425 \cdot 3 + 1.15 = 0.325$



Ejercicio. Estimar la cota del error usando la expresión (16).

Aproximación de Funciones

Polinomio interpolante de Lagrange

Una dificultad que surge al usar estos polinomios es que como es difícil trabajar con el término de error dado por (16), no se sabe generalmente el grado del polinomio necesario para lograr la precisión deseada hasta que los cálculos han sido completados.

La práctica usual consiste en comparar los resultados obtenidos de varios polinomios hasta que se obtiene una concordancia apropiada.

Veremos ahora la derivación de estos polinomios aproximantes de tal manera que se utilicen con mayor ventaja los cálculos de los polinomios anteriores para obtener el nuevo polinomio aproximante.

Aproximación de Funciones

Polinomio interpolante de Lagrange

Sea f una función definida en x_0, x_1, \dots, x_n , y supongamos que m_1, m_2, \dots, m_k son k enteros distintos con $0 \leq m_i \leq n$.

El polinomio de Lagrange de grado menor que k que coincide con f en los puntos $x_{m_1}, x_{m_2}, \dots, x_{m_k}$ se denota por $P_{m_1, m_2, \dots, m_k}(x)$.

Teorema. Sea f definida en x_0, x_1, \dots, x_k y sean x_j y x_i dos números distintos de este conjunto. Si

$$p(x) = \frac{(x - x_j)P_{0,1,\dots,j-1,j+1,\dots,k}(x) - (x - x_i)P_{0,1,\dots,i-1,i+1,\dots,k}(x)}{x_i - x_j},$$

entonces $p(x)$ es el polinomio de Lagrange de grado menor o igual a k , que interpola a f en x_0, x_1, \dots, x_k .

Obs. Este resultado nos da un método para generar en forma recursiva aproximaciones polinómicas de Lagrange.

Aproximación de Funciones

Polinomio interpolante de Lagrange

Ejemplo. Supongamos que los valores de una función f están dados en los puntos x_0, x_1, \dots, x_4 , podemos construir aproximaciones de la función f en un punto x a partir de la información dada.

Así, $P_0 = f(x_0)$, $P_1 = f(x_1)$, $P_2 = f(x_2)$, $P_3 = f(x_3)$, $P_4 = f(x_4)$

$$P_{0,1}(x) = \frac{(x - x_0)P_1(x) - (x - x_1)P_0(x)}{x_1 - x_0}, \quad P_{1,2}(x) = \frac{(x - x_1)P_2(x) - (x - x_2)P_1(x)}{x_2 - x_1}, \quad \dots$$

es decir, se tienen los polinomios siguientes, evaluados en el punto x

x_0	P_0					
x_1	P_1	$P_{0,1}$				
x_2	P_2	$P_{1,2}$	$P_{0,1,2}$			
x_3	P_3	$P_{2,3}$	$P_{1,2,3}$	$P_{0,1,2,3}$		
x_4	P_4	$P_{3,4}$	$P_{2,3,4}$	$P_{1,2,3,4}$	$P_{0,1,2,3,4}$	

$$P_{0,1,2}(x) = \frac{(x - x_0)P_{1,2}(x) - (x - x_2)P_{0,1}(x)}{x_2 - x_0},$$

De aquí, si no estamos conforme con la aproximación $P_{0,1,2,3,4}$ en x para f ,

podemos seleccionar otro punto x_5 , se calcula otra fila a la tabla y se

comparan $P_{0,1,2,3,4}$, $P_{1,2,3,4,5}$ y $P_{0,1,2,3,4,5}$ para determinar la precisión adicional.

Aproximación de Funciones

Polinomio interpolante de Lagrange

El procedimiento descrito se conoce como el **método de Neville**.

Leer x, x_0, x_1, \dots, x_n y $f(x_0), f(x_1), \dots, f(x_n)$

Almacenar los valores de f como la primera columna de la matriz Q ($Q_{0,0}$ a $Q_{n,0}$)

Para $i = 1$ hasta n

Para $j = 1$ hasta i

$$Q_{i,j} = \frac{(x - x_{i-j})Q_{i,j-1} - (x - x_i)Q_{i-1,j-1}}{x_i - x_{i-j}}$$

Fin para

Fin para

Modificar el algoritmo para permitir agregar nuevos puntos interpolantes.

$$Q_{2,2} = \frac{(x - x_0)Q_{2,1} - (x - x_2)Q_{1,1}}{x_2 - x_0}$$

x_0	$Q_{0,0}$				
x_1	$Q_{1,0}$	$Q_{1,1}$			
x_2	$Q_{2,0}$	$Q_{2,1}$	$Q_{2,2}$		
x_3	$Q_{3,0}$	$Q_{3,1}$	$Q_{3,2}$	$Q_{3,3}$	
x_4	$Q_{4,0}$	$Q_{4,1}$	$Q_{4,2}$	$Q_{4,3}$	$Q_{4,4}$

Aproximación de Funciones

Interpolación polinómica

El problema de interpolación no finaliza con la determinación del polinomio interpolante. En muchas aplicaciones debemos evaluar el polinomio en puntos donde no se conoce el valor de la función.

Hemos visto, que determinar el polinomio interpolante de Lagrange $p(x)$ es muy sencillo, ya que sus coeficientes son los valores de la función $f(x_i)$

$$p(x) = f(x_0)L_0(x) + f(x_1)L_1(x) + \cdots + f(x_n)L_n(x) = \sum_{i=0}^n f(x_i)L_i(x),$$

pero evaluar los polinomios individuales de Lagrange en un punto x , es decir $L_i(x)$, no lo es tan fácil, ya que envuelve productos de la forma

$$(x_j - x_0) \cdots (x_j - x_{j-1})(x_j - x_{j+1}) \cdots (x_j - x_n)$$

y este cálculo puede producir fácilmente overflow o underflow.

Aproximación de Funciones

Interpolación polinómica

A pesar de que los coeficientes del polinomio interpolante natural

$$p(t) = a_n t^n + a_{n-1} t^{n-1} + \dots + a_2 t^2 + a_1 t + a_0, \quad (17)$$

no son fáciles de calcular, éste puede ser evaluado eficiente y establemente mediante el algoritmo denominado **evaluación anidada** (método de Horner)

Para esto, reescribimos el polinomio (17) en la forma

$$p(t) = ((\dots((a_n t + a_{n-1}) t + a_{n-2}) \dots) t + a_1) t + a_0. \quad (18)$$

Probar que (17) y (18) son equivalentes usando inducción,

La forma (18) sugiere
la evaluación sucesiva
siguiente

$$\left\{ \begin{array}{l} a_n, \\ (a_n t + a_{n-1}), \\ ((a_n t + a_{n-1}) t + a_{n-2}), \\ \dots \\ (\dots((a_n t + a_{n-1}) t + a_{n-2}) \dots) t + a_1, \\ ((\dots((a_n t + a_{n-1}) t + a_{n-2}) \dots) t + a_1) t + a_0 \end{array} \right.$$

Aproximación de Funciones

Interpolación polinómica

Esta idea conduce al algoritmo siguiente:

$$\left\{ \begin{array}{l} p = a_n \\ \text{Para } i = n-1 \text{ hasta } 0 \\ \quad p = p * t + a_i \\ \text{Fin para} \end{array} \right.$$

Al final, p contiene la evaluación del polinomio en el punto t . Este algoritmo es bastante eficiente, sólo requiere de n sumas y n multiplicaciones.

Obs. Tenemos:

- el polinomio interpolante natural es difícil de calcular, pero es fácil de evaluar
- el polinomio interpolante de Lagrange es fácil de determinar, pero difícil de evaluar

La pregunta que surge es:

¿Existe un polinomio interpolante que sea fácil de determinar y fácil de evaluar? La respuesta es si.

Aproximación de Funciones

Polinomio interpolante de Newton

Este toma como base los polinomios

$$1, (x - x_0), (x - x_0)(x - x_1), \dots, (x - x_0)(x - x_1) \cdots (x - x_{n-1}) \quad (19)$$

o de manera equivalente, el polinomio interpolante es

$$p(x) = c_0 + c_1(x - x_0) + c_2(x - x_0)(x - x_1) + \dots \\ + c_n(x - x_0)(x - x_1) \cdots (x - x_{n-1}) \quad (20)$$

Veamos como evaluar un polinomio de este tipo. Para esto escribimos el polinomio (20) usando una forma anidada

$$p(x) = ((\cdots((c_n)(x - x_{n-1}) \\ + c_{n-1})(x - x_{n-2}) + c_{n-2}) \cdots)(x - x_1) + c_1)(x - x_0) + c_0,$$

de donde el algoritmo para evaluar p en un punto t es

$$\left\{ \begin{array}{l} p = c_n \\ \text{Para } i = n-1 \text{ hasta } 0 \\ \quad p = p * (t - x_i) + c_i \\ \text{Fin para} \end{array} \right.$$

Requiere:
2n sumas/restas y
n multiplicaciones.

Aproximación de Funciones

Polinomio interpolante de Newton

Para determinar el polinomio evaluamos (20) en los puntos x_i ,

$$p(x_0) = c_0 + c_1(x_0 - x_0) + c_2(x_0 - x_0)(x_0 - x_1) + \dots \\ + c_n(x_0 - x_0)(x_0 - x_1) \cdots (x_0 - x_{n-1}) = c_0$$

$$p(x_1) = c_0 + c_1(x_1 - x_0) + c_2(x_1 - x_0)(x_1 - x_1) + \dots \\ + c_n(x_1 - x_0)(x_1 - x_1) \cdots (x_1 - x_{n-1}) = c_0 + c_1(x_1 - x_0)$$

En general, $p(x_i)$ contendrá sólo $i+1$ términos no cero, ya que los últimos $n-i$ términos contienen el factor $(x - x_i)$ evaluado en $x = x_i$.

Usando las condiciones de interpolación $f(x_i) = p(x_i)$, podemos determinar los coeficientes c_i del sistema siguiente

$$\begin{aligned} f(x_0) &= c_0, \\ f(x_1) &= c_0 + c_1(x_1 - x_0), \\ &\dots \\ f(x_n) &= c_0 + c_1(x_n - x_0) + \dots + c_n(x_n - x_0) \cdots (x_n - x_{n-1}) \end{aligned} \tag{21}$$

Aproximación de Funciones

Polinomio interpolante de Newton

La matriz del sistema (21) es triangular inferior, con elementos en la diagonal distintos de cero, y por lo tanto no singular

$$\begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 1 & (x_1 - x_0) & 0 & \cdots & 0 \\ 1 & (x_2 - x_0) & (x_2 - x_0)(x_2 - x_1) & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & (x_n - x_0) & (x_n - x_0)(x_n - x_1) & \cdots & (x_n - x_0)(x_n - x_1) \cdots (x_n - x_{n-1}) \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ \vdots \\ f_n \end{pmatrix} \quad (22)$$

Obs. Una consecuencia interesante de la triangularidad del sistema (22) es que la adición de nuevos puntos al problema de interpolación no afecta a los coeficientes que se han calculado.

Obs. Notar que

c_0	polinomio de grado cero interpolante en (x_0, f_0)
$c_0 + c_1(x - x_0)$,	polinomio de grado uno interpolante en $(x_0, f_0), (x_1, f_1)$
$c_0 + c_1(x - x_0) + c_2(x - x_0)(x - x_1)$	polinomio de grado dos interpolante en $(x_0, f_0), (x_1, f_1), (x_2, f_2)$
...	

Aproximación de Funciones

Polinomio interpolante de Newton

Obs. En principio el sistema triangular (22) puede ser resuelto en $O(n^2)$ operaciones para calcular los coeficientes c_i del polinomio interpolante de Newton. Desafortunadamente, las entradas de la matriz del sistema pueden producir overflow o underflow fácilmente.

Tomaremos otro tipo de enfoque que nos permita determinar los coeficientes c_i del polinomio interpolante de Newton de manera de evitar este inconveniente.

Aproximación de Funciones

Polinomio interpolante de Newton

A partir del sistema (22) se tiene para los 2 primeras ecuaciones

$$c_0 = f(x_0) \quad \text{y} \quad c_1 = \frac{f(x_1) - f(x_0)}{x_1 - x_0}.$$

Introducimos lo que se conoce como notación de **diferencia dividida**.

La diferencia dividida cero para la función f con respecto a x_i es

$$f[x_i] = f(x_i)$$

La diferencia dividida uno para la función f con respecto a x_i y x_{i+1} es

$$f[x_i, x_{i+1}] = \frac{f[x_{i+1}] - f[x_i]}{x_{i+1} - x_i}. \quad (23)$$

La diferencia dividida dos para la función f con respecto a x_i , x_{i+1} y x_{i+2} es

$$f[x_i, x_{i+1}, x_{i+2}] = \frac{f[x_{i+1}, x_{i+2}] - f[x_i, x_{i+1}]}{x_{i+2} - x_i}. \quad (24)$$

Las diferencias divididas restantes se definen inductivamente.

Aproximación de Funciones

Polinomio interpolante de Newton

Cuando las $k-1$ diferencias divididas

$$f[x_i, \dots, x_{i+k-1}] \quad \text{y} \quad f[x_{i+1}, \dots, x_{i+k}]$$

han sido determinadas, la k -ésima diferencia dividida de f respecto $x_i, x_{i+1}, \dots, x_{i+k}$, esta dada por

$$f[x_i, \dots, x_{i+k}] = \frac{f[x_{i+1}, \dots, x_{i+k}] - f[x_i, \dots, x_{i+k-1}]}{x_{i+k} - x_i}. \quad (25)$$

Usando esta notación, los otros coeficientes del polinomio interpolante (20), es decir c_2 hasta c_n , se pueden obtener consecutivamente de una manera similar a c_0 y c_1 . Así,

$$c_0 = f[x_0], \quad c_1 = f[x_0, x_1], \quad \dots, \quad c_k = f[x_0, \dots, x_k]. \quad (26)$$

Ejercicio. Dado el polinomio interpolante (20), con c_0 y c_1 como en (26), usar $p(x_2)$ para demostrar que $c_2 = f[x_0, x_1, x_2]$.

Aproximación de Funciones

Polinomio interpolante de Newton

Con la notación (26), el polinomio interpolante dado por la ecuación (20) se puede escribir como

$$\begin{aligned}
 p(x) &= f[x_0] + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) + \dots \\
 &\quad + f[x_0, \dots, x_n](x - x_0)(x - x_1) \cdots (x - x_{n-1}) \\
 p(x) &= f[x_0] + \sum_{k=1}^n f[x_0, \dots, x_k] \prod_{j=0}^{k-1} (x - x_j)
 \end{aligned} \tag{27}$$

con x_0, x_1, \dots, x_n , puntos distintos.

La ecuación (27) se conoce como la fórmula de **diferencia dividida interpolante de Newton**. Esta nos da un procedimiento iterado para calcular los coeficientes del polinomio interpolante $p(x)$.

La determinación de las diferencias divididas para puntos de datos tabulados se bosqueja en la siguiente tabla.

Aproximación de Funciones

$$f[x_i, \dots, x_{i+k}] = \frac{f[x_{i+1}, \dots, x_{i+k}] - f[x_i, \dots, x_{i+k-1}]}{x_{i+k} - x_i}$$

Polinomio interpolante de Newton

	diferencia dividida					
	cero	uno	dos	tres	cuatro	cinco
x_0	$f[x_0]$					
x_1	$f[x_1]$	$f[x_0, x_1]$				
x_2	$f[x_2]$	$f[x_1, x_2]$	$f[x_0, x_1, x_2]$			
x_3	$f[x_3]$	$f[x_2, x_3]$	$f[x_1, x_2, x_3]$	$f[x_0, x_1, x_2, x_3]$		
x_4	$f[x_4]$	$f[x_3, x_4]$	$f[x_2, x_3, x_4]$	$f[x_1, x_2, x_3, x_4]$	$f[x_0, x_1, x_2, x_3, x_4]$	
x_5	$f[x_5]$	$f[x_4, x_5]$	$f[x_3, x_4, x_5]$	$f[x_2, x_3, x_4, x_5]$	$f[x_1, x_2, x_3, x_4, x_5]$	$f[x_0, x_1, x_2, x_3, x_4, x_5]$

Los elementos de la diagonal son los coeficientes del polinomio (27),

$$p(x) = f[x_0] + \sum_{k=1}^5 f[x_0, \dots, x_k] \prod_{j=0}^{k-1} (x - x_j).$$

Este método permite agregar punto adicionales a bajo costo computacional.

Aproximación de Funciones

Polinomio interpolante de Newton

Algoritmo para calcular los coeficientes del polinomio

(El número de operaciones para este algoritmo es n^2 sumas y $n^2/2$ divisiones)

Leer $x_i, f_i = f(x_i), i = 1, \dots, n$

Almacenar f_i en $c(i), i = 1, \dots, n$

Para $j = 2$ hasta n

Para $i = n$ hasta j

$$c(i) = (c(i) - c(i - 1)) / (x(i) - x(i - j + 1))$$

Fin para

Fin para

No es necesario almacenar todo el arreglo bidimensional, de la tabla anterior. Durante los cálculos el arreglo c se sobre-escribe, y al final contendrá los coeficientes del polinomio.

Ejercicio: Extender el algoritmo para almacenar todas la diferencias divididas.

Obs. La forma de diferencia dividida interpolante de Newton nos permite construir el polinomio de interpolación (20), y el algoritmo de evaluación anidada nos ayuda a evaluar este polinomio en puntos no tabulados.

Aproximación de Funciones

Polinomio interpolante de Newton

Ejemplo:

i	x_i	diferencia dividida				
		cero	uno	dos	tres	cuatro
1	1.0000	0.7651977				
2	1.3000	0.6200860	-0.4837			
3	1.6000	0.4554022	-0.5489	-0.1087		
4	1.9000	0.2818186	-0.5786	-0.0494	0.0659	
5	2.2000	0.1103623	-0.5715	0.0118	0.0681	0.0018

diferencia dividida $f[x_i, \dots, x_{i+k}] = \frac{f[x_{i+1}, \dots, x_{i+k}] - f[x_i, \dots, x_{i+k-1}]}{x_{i+k} - x_i}$.

$$p(x) = 0.7652 - 0.4837(x - x_1) - 0.1087(x - x_1)(x - x_2) + 0.0659(x - x_1)(x - x_2)(x - x_3) + 0.0018(x - x_1)(x - x_2)(x - x_3)(x - x_4)$$

Aproximación de Funciones

Polinomio interpolante de Newton

Teorema. Si $f \in C^n[a,b]$ y $x_0, x_1, \dots, x_n \in [a,b]$ son distintos, entonces existe un número ξ en $[a,b]$ tal que

$$f[x_0, \dots, x_n] = \frac{f^{(n)}(\xi)}{n!}.$$

Obs. Este resultado es una generalización de la aplicación del teorema del valor medio a

$$f[x_0, x_1] = \frac{f[x_1] - f[x_0]}{x_1 - x_0},$$

es decir, si f' existe, $f[x_0, x_1] = f'(\xi)$ para algún número ξ entre x_0 y x_1 .

Teorema (término residual o cota de error). Sea p el polinomio interpolante de f en los puntos x_0, x_1, \dots, x_n (puntos diferentes) en $[a,b]$, dado por (27).

Entonces para cada $x \in [a,b]$ existe un número $\xi(x)$ en (a,b) tal que

$$f(x) - p(x) = \frac{f^{(n+1)}(\xi(x))}{(n+1)!} (x - x_0) \cdots (x - x_n). \quad (28)$$

Aproximación de Funciones

Polinomio interpolante de Newton

Veamos la manera de cómo usar la acotación del error dado por (28) en un caso sencillo. Sea $l(x)$ el polinomio lineal interpolante de $f(x)$ en x_0 y x_1 y supongamos además que

$$|f''(x)| \leq M$$

en el intervalo de interés. Entonces, de (28)

$$|f(x) - l(x)| = \frac{|f''(\xi)|}{2} |(x - x_0)(x - x_1)| \leq \frac{M}{2} |(x - x_0)(x - x_1)|. \quad (29)$$

El problema del estudio de esta acotación depende en que x este fuera o dentro del intervalo $[x_0, x_1]$.

Si x esta fuera de $[x_0, x_1]$, se dice que estamos extrapolando para aproximar a f . Como $|(x - x_0)(x - x_1)|$ crece rápidamente a medida que x se aleja del intervalo $[x_0, x_1]$, extrapolación es un riesgo.

Si x esta dentro de $[x_0, x_1]$, se dice que estamos interpolando para aproximar f .

Aproximación de Funciones

Polinomio interpolante de Newton

En el caso de interpolación, podemos obtener cotas de error uniforme para (29). La función $|(x-x_0)(x-x_1)|$ alcanza su máximo en el punto $x = (x_0+x_1)/2$ siendo su máximo valor $(x_1-x_0)^2/4$. De aquí

$$x \in [x_0, x_1] \Rightarrow |f(x) - l(x)| \leq \frac{M}{2} |(x-x_0)(x-x_1)| \leq \frac{M}{8} (x_1-x_0)^2. \quad (30)$$

Como una aplicación, supongamos que queremos calcular valores de la función $\sin(x)$ (rápido y fácil) almacenando valores de esta función en puntos equidistantes (distancia h) y usando interpolación lineal.

La pregunta que surge es: por ejemplo para un error de 10^{-4}

¿Cuán pequeño debe ser h para alcanzar una precisión dada?

Como la segunda derivada de $\sin(x)$ está acotada por 1, se tiene de (30)

$$|\sin(x) - l(x)| \leq \frac{h^2}{8}.$$

Tomando $\frac{h^2}{8} \leq 10^{-4}$ se tiene $h \leq 0.01\sqrt{8} = 0.0283\dots$

Aproximación de Funciones

Polinomio interpolante de Newton

El método descrito para aproximar el $\sin(x)$ usa muchos puntos sobre intervalos pequeños (h es muy pequeño).

Otra posibilidad, es usar polinomios interpolantes de grado más alto para representar la función f .

Supongamos que para cada $n = 1, 2, 3, \dots$, escogemos $n+1$ puntos equiespaciados y sea p_n el polinomio interpolante de f en esos puntos.

Si la sucesión de polinomios converge uniformemente a f , sabemos que existe un número n para el cual p_n está suficientemente de cerca de f para una precisión dada.

Veamos el siguiente ejemplo.

Aproximación de Funciones

Polinomio interpolante natural

Ejemplo: $(-2, -27), (0, -1), (1, 0)$.

$$p_2(t) = x_1 + x_2t + x_3t^2$$

$$\begin{pmatrix} 1 & t_1 & t_1^2 \\ 1 & t_2 & t_2^2 \\ 1 & t_3 & t_3^2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix}$$

$$\begin{pmatrix} 1 & -2 & 4 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} -27 \\ -1 \\ 0 \end{pmatrix}$$

Solución: $x = (-1, 5, -4)^t$. Entonces,

$$p_2(t) = -1 + 5t - 4t^2.$$

Aproximación de Funciones

Polinomio interpolante de Lagrange

Ejemplo: $(-2, -27), (0, -1), (1, 0)$.

El polinomio $p_2(t)$ es

$$y_1 \frac{(t - t_2)(t - t_3)}{(t_1 - t_2)(t_1 - t_3)} + y_2 \frac{(t - t_1)(t - t_3)}{(t_2 - t_1)(t_2 - t_3)} + y_3 \frac{(t - t_1)(t - t_2)}{(t_3 - t_1)(t_3 - t_2)}$$

$$p_2(t) = -27 \frac{t(t - 1)}{6} + \frac{(t + 2)(t - 1)}{2}.$$

Note: es el mismo polinomio hallado con la base de monomios, $p_2(t) = -1 + 5t - 4t^2$.

Aproximación de Funciones

Polinomio interpolante de Newton

Ejemplo: $(-2, -27), (0, -1), (1, 0)$.

t_i	$f[t_i]$	$f[t_i, t_{i+1}]$	$f[t_i, t_{i+1}, t_{i+2}]$
-2	-27		
0	-1	$\frac{-1 - (-27)}{0 - (-2)} = 13$	
1	0	$\frac{0 - (-1)}{1 - 0} = 1$	$\frac{1 - 13}{1 - (-2)} = -4$

Solución:

$$p_2(t) = -27 + 13(t + 2) - 4(t + 2)t$$

$$p_2(t) = -1 + 5t - 4t^2.$$

Aproximación de Funciones

Polinomio interpolante de Newton

Consideremos la función $f(x) = \frac{1}{1+25x^2}$ para $x \in [-1,1]$.

Seleccionamos los puntos

$$x_i = -1 + i \frac{2}{n}$$

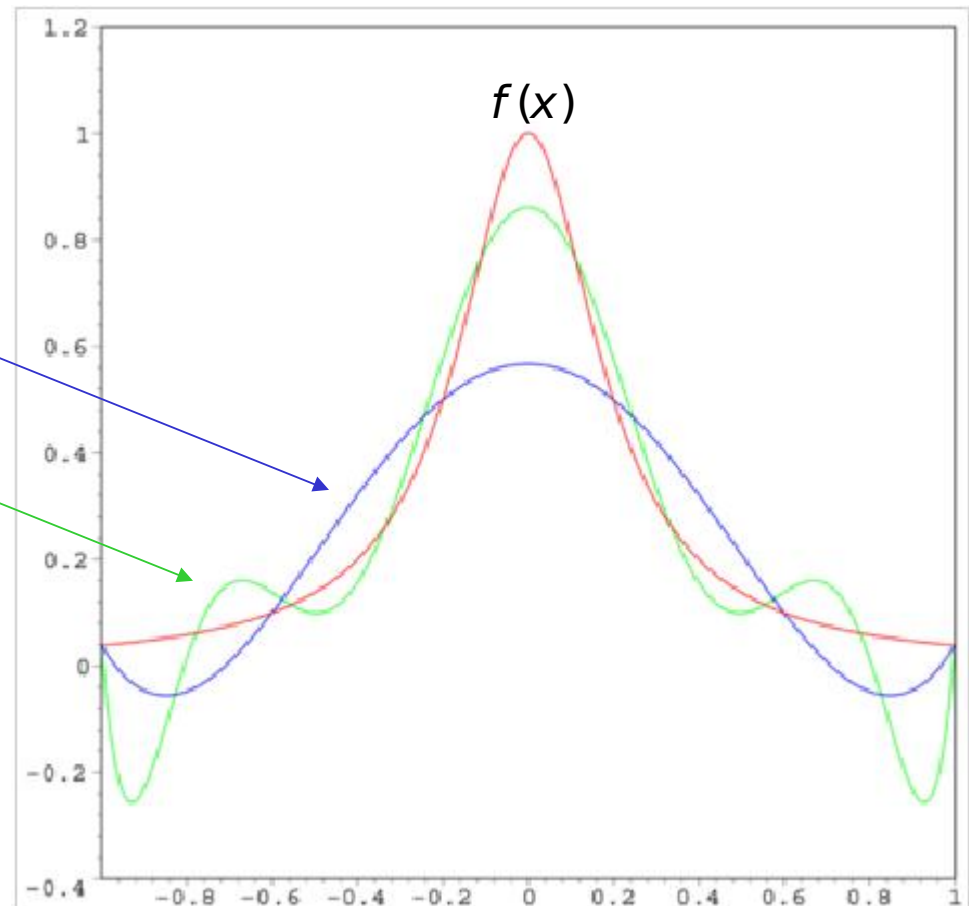
para $i \in \{0,1,\dots,n\}$

Polinomio interpolante de grado 5
(construido con 6 puntos)

Polinomio interpolante de grado 9
(construido con 10 puntos)

El error de interpolación tiende a infinito cuando el grado del polinomio crece:

$$\lim_{n \rightarrow \infty} (\max_{-1 \leq x \leq 1} |f(x) - p_n(x)|) = \infty$$



Aproximación de Funciones

Polinomio interpolante de Newton

Para la función $f(x) = \frac{1}{1+25x^2}$ para $x \in [-1,1]$.

sus dos primeras derivadas son

$$f'(x) = -\frac{50x}{(1+25x^2)^2} \Rightarrow |f'(1)| = \frac{50}{26^2} = 0.0740$$

$$f''(x) = -\frac{5000(1+25x^2) - 50(1+25x^2)^2}{(1+25x^2)^4} \Rightarrow |f''(1)| = \frac{96200}{26^4} = 0.2105$$

La magnitud de las derivadas de orden alto para esta función crecen más.

Vimos que $f(x) - p(x) = \frac{f^{(n+1)}(\xi(x))}{(n+1)!} (x-x_0) \cdots (x-x_n)$,

por lo tanto, la cota del error de interpolación cuando se usan polinomios de grado alto crece mucho.

Aproximación de Funciones

Puntos de interpolación de Chebyshev

Para la función $f(x) = \frac{1}{1+25x^2}$ con $x \in [-1,1]$,

vimos que el error de interpolación tiende a infinito cuando el grado del polinomio crece. Este hecho se conoce como el fenómeno de Runge.

Los puntos de Chebyshev surgen del esfuerzo para ajustar los puntos de interpolación y tratar controlar el error de interpolación.

Dado m , los m puntos de Chebyshev en el intervalo $[-1,1]$, es decir x_0, x_1, x_{m-1} se definen como

$$x_{i-1} = \cos\left(\frac{2i-1}{2m}\pi\right), \quad 1 \leq i \leq m. \quad (31)$$

Para definir m puntos de Chebyshev en un intervalo dado $[a,b]$ se procede así

$$x_{i-1} = \frac{1}{2}(a+b) + \frac{1}{2}(b-a)\cos\left(\frac{2i-1}{2m}\pi\right), \quad 1 \leq i \leq m. \quad (32)$$

Aproximación de Funciones

Puntos de interpolación de Chebyshev

Caso 5 puntos en $[-1, 1]$

puntos igualmente espaciados

i	x_i	diferencia dividida				
		cero	uno	dos	tres	cuatro
0	-1.0000	0.0385				
1	-0.5000	0.1379	0.1989			
2		1.0000	1.7241	1.5252		
3	+0.0000	0.1379	-1.7241	-3.4483	-3.3156	
4	+0.5000	0.0385	-0.1989	1.5252	3.3156	3.3156

polinomio interpolante de Newton

$$p(x) = 0.0385 + 0.1989(x - x_0) + 1.5252(x - x_0)(x - x_1) - 3.3156(x - x_0)(x - x_1)(x - x_2) + 3.3156(x - x_0)(x - x_1)(x - x_2)(x - x_3)$$

$$f[x_i, \dots, x_{i+k}] = \frac{f[x_{i+1}, \dots, x_{i+k}] - f[x_i, \dots, x_{i+k-1}]}{x_{i+k} - x_i}$$

Aproximación de Funciones

Puntos de interpolación de Chebyshev

Caso 5 puntos en $[-1, 1]$

puntos de Chebyshev

i	x_i	diferencia dividida				
		cero	uno	dos	tres	cuatro
0	-0.9511	0.0424				
1	-0.5878	0.1038	0.1691			
2	0.0000	1.0000	1.5248	1.4255		
3	+0.5878	0.1038	-1.5248	-2.5941	-2.6121	
4	+0.9511	0.0424	-0.1691	1.4255	2.6121	2.7465

polinomio interpolante de Newton

$$p(x) = 0.0424 + 0.1691(x - x_0) + 1.4255(x - x_0)(x - x_1) - 2.6121(x - x_0)(x - x_1)(x - x_2) + 2.7465(x - x_0)(x - x_1)(x - x_2)(x - x_3)$$

$$f[x_i, \dots, x_{i+k}] = \frac{f[x_{i+1}, \dots, x_{i+k}] - f[x_i, \dots, x_{i+k-1}]}{x_{i+k} - x_i}$$

Aproximación de Funciones

Caso 5 puntos en $[-1, 1]$

puntos igualmente espaciados

-1.0000	-0.5000	0.0000	0.5000	1.0000
0.0385	0.1379	1.0000	0.1379	0.0385

puntos de Chebyshev

-0.9511	-0.5878	0.0000	0.5878	0.9511
0.0424	0.1038	1.0000	0.1038	0.0424

Polinomio interpolante de Newton

$$f(x) = \frac{1}{1+25x^2} \quad \text{para } x \in [-1,1].$$

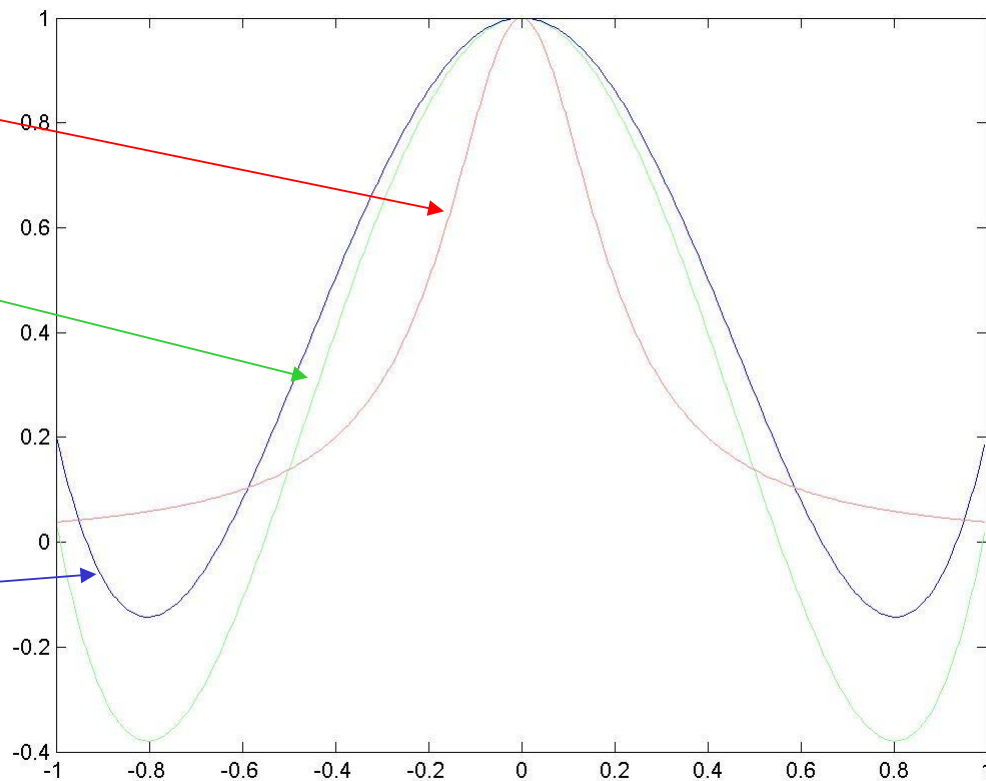
Coeficientes polinomio interpolante de Newton de grado 4 para puntos igualmente espaciados

0.0385	0.1989	1.5252	-3.3156	3.3156
--------	--------	--------	---------	--------

Coeficientes polinomio interpolante de Newton de grado 4 para puntos de Chebyshev

0.0424	0.1691	1.4255	-2.6121	2.7465
--------	--------	--------	---------	--------

El error de interpolación es menor



Aproximación de Funciones

Puntos de interpolación de Chebyshev

El error de interpolación para el polinomio interpolante de Newton en $n+1$ puntos es

$$f(x) - p(x) = \frac{f^{(n+1)}(\xi(x))}{(n+1)!} (x - x_0) \cdots (x - x_n).$$

Si suponemos que $|f^{(n+1)}(x)| \leq M$ si $-1 \leq x \leq 1$, entonces

$$|f(x) - p(x)| \leq \frac{M}{(n+1)!} \max_{x \in [-1,1]} \left| \prod_{i=0}^n (x - x_i) \right|.$$

Se puede demostrar que variando los puntos x_i , $0 \leq i \leq n$

$$\min \left(\max_{x \in [-1,1]} \left| \prod_{i=0}^n (x - x_i) \right| \right) = 2^{-n}$$

y este mínimo se alcanza en los puntos de Chebyshev definidos por (31).

Aproximación de Funciones

Interpolación polinómica a trozos

Hasta el momento se ha realizado aproximación de funciones arbitrarias en intervalos cerrados usando polinomios. Este método es apropiado en muchas circunstancias, pero

- la naturaleza oscilatoria de los polinomios de grado alto y
- la propiedad de que una fluctuación sobre una porción pequeña del intervalo puede introducir fluctuaciones muy grandes sobre el rango entero,

restringe su uso cuando se aproximan muchas de las funciones que surgen en situaciones físicas.

Un enfoque alternativo es dividir el intervalo en una colección de subintervalos y construir un polinomio interpolación diferente en cada subintervalo. La aproximación con funciones de este tipo se denomina interpolación polinómica a trozos (“splines”).

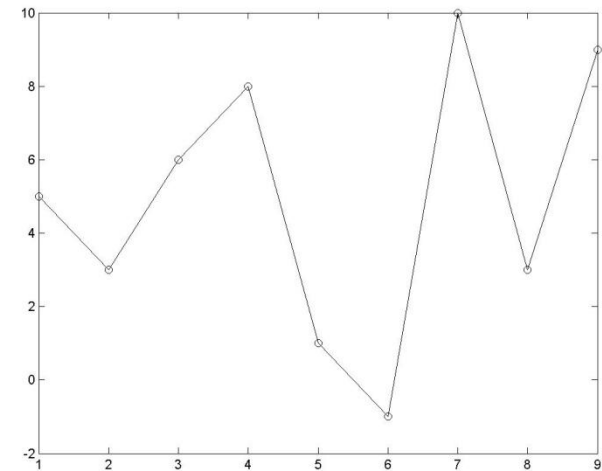
Aproximación de Funciones

Interpolación polinómica a trozos

El tipo más simple de interpolación polinómica a trozos es la lineal a trozos que consiste en unir un conjunto de $n+1$ puntos

$$\{(x_0, f(x_0)), (x_1, f(x_1)), \dots, (x_n, f(x_n))\}$$

con una serie de líneas rectas. Este es un método usado, por ejemplo, para funciones trigonométricas, cuando se quieren los valores intermedios de una colección de puntos tabulados.



La desventaja de enfocar un problema de interpolación usando funciones lineales es que en cada uno de los extremos de los subintervalos, no hay ninguna seguridad de diferenciabilidad, lo cual, geoméricamente significa que la función interpolante no es “suave” en esos puntos.

Aproximación de Funciones

Interpolación polinómica a trozos

Definición. Dada una función f definida en $[a, b]$ y un conjunto de números, los cuales denominaremos los nodos, $a = x_0 < x_1 < \dots < x_n = b$, un **spline cúbico** S para f es una función que satisface las condiciones

a) S es un polinomio de grado ≤ 3 , denotado por S_j en el intervalo

$[x_j, x_{j+1}]$ para cada $j = 0, 1, \dots, n-1$

b) $S(x_j) = f(x_j)$ para cada $j = 0, 1, \dots, n$

c) $S'_{j+1}(x_{j+1}) = S'_j(x_{j+1})$ para cada $j = 0, 1, \dots, n-2$

d) $S''_{j+1}(x_{j+1}) = S''_j(x_{j+1})$ para cada $j = 0, 1, \dots, n-2$

e) se satisface una del siguiente conjunto de condiciones de frontera

i. $S''(x_0) = S''(x_n) = 0$ (frontera libre)

ii. $S'(x_0) = f'(x_0)$ y $S'(x_n) = f'(x_n)$ (frontera amarrada)

En el caso (i) S se denomina spline cúbico natural.

Aproximación de Funciones

Interpolación polinómica a trozos

Para construir el spline cúbico para una función f , aplicamos las condiciones de la definición al polinomio cúbico

$$S_j(x) = a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3 \quad \text{para cada } j = 0, \dots, n-1.$$

Claramente $S_j(x_j) = a_j = f(x_j)$,

y aplicando la condición (b), se tiene que para $j = 0, \dots, n-2$

$$a_{j+1} = S_{j+1}(x_{j+1}) = S_j(x_{j+1}) = a_j + b_j(x_{j+1} - x_j) + c_j(x_{j+1} - x_j)^2 + d_j(x_{j+1} - x_j)^3.$$

Introducimos la notación $h_j = (x_{j+1} - x_j)$ para cada $j = 0, \dots, n-1$,

y si definimos $a_n = f(x_n)$ se tiene que

$$a_{j+1} = a_j + b_j h_j + c_j h_j^2 + d_j h_j^3 \quad \text{para cada } j = 0, \dots, n-1. \quad (33)$$

Aproximación de Funciones

Interpolación polinómica a trozos

Si definimos $b_n = S'(x_n)$ y observamos que

$$S'_j(x) = b_j + 2c_j(x - x_j) + 3d_j(x - x_j)^2$$

se tiene que $S'_j(x_j) = b_j$ para cada $j = 0, \dots, n-1$.

Aplicando la condición (c)

$$b_{j+1} = b_j + 2c_j h_j + 3d_j h_j^2 \quad \text{para cada } j = 0, \dots, n-1. \quad (34)$$

Otra relación entre los coeficientes de S_j se obtiene definiendo

$$c_n = S''_{n-1}(x_n) / 2$$

y aplicando la condición (d), de donde

$$c_{j+1} = c_j + 3d_j h_j \quad \text{para cada } j = 0, \dots, n-1. \quad (35)$$

Aproximación de Funciones

Interpolación polinómica a trozos

Despejando d_j de (35)

$$d_j = \frac{c_{j+1} - c_j}{3h_j} \quad (36)$$

y sustituyendo en (33) y (34) se tiene que para cada $j = 0, \dots, n-1$

$$a_{j+1} = a_j + b_j h_j + \frac{h_j^2}{3} (2c_j + c_{j+1}) \quad (37)$$

$$b_{j+1} = b_j + h_j (c_j + c_{j+1}) \quad (38)$$

Despejando b_j en la ecuación (37)

$$b_j = \frac{1}{h_j} (a_{j+1} - a_j) - \frac{h_j}{3} (2c_j + c_{j+1}) \quad (39)$$

y luego b_{j-1} en la misma ecuación (con una reducción del índice)

$$b_{j-1} = \frac{1}{h_{j-1}} (a_j - a_{j-1}) - \frac{h_{j-1}}{3} (2c_{j-1} + c_j). \quad (40)$$

Aproximación de Funciones

Interpolación polinómica a trozos

Reduciendo el índice en 1 para (38) se tiene $b_j = b_{j-1} + h_{j-1}(c_{j-1} + c_j)$.

Sustituyendo b_j y b_{j-1} dados por (39) y (40), en la ecuación anterior

$$\frac{1}{h_j}(a_{j+1} - a_j) - \frac{h_j}{3}(2c_j + c_{j+1}) = \frac{1}{h_{j-1}}(a_j - a_{j-1}) - \frac{h_{j-1}}{3}(2c_{j-1} + c_j) + h_{j-1}(c_{j-1} + c_j)$$

Realizando algunas simplificaciones se obtiene el sistema lineal

$$h_{j-1}c_{j-1} + 2(h_{j-1} + h_j)c_j + h_jc_{j+1} = \frac{3}{h_j}(a_{j+1} - a_j) - \frac{3}{h_{j-1}}(a_j - a_{j-1})$$

para cada $j = 1, \dots, n - 1$,

que escrito en forma matricial queda como

Aproximación de Funciones

Interpolación polinómica a trozos

Una vez determinados los c_j , los b_j se calculan usando (39) y los d_j de (36)

$$b_j = \frac{1}{h_j} (a_{j+1} - a_j) - \frac{h_j}{3} (2c_j + c_{j+1}) \quad d_j = \frac{c_{j+1} - c_j}{3h_j}$$

para cada $j = 0, \dots, n - 1$.

Finalmente se tiene los coeficientes del spline cúbico S_j en cada subintervalo $[x_j, x_{j+1}]$

$$S_j(x) = a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3$$

para cada $j = 0, \dots, n - 1$.

La pregunta que surge es ¿cuándo el sistema lineal (41) tiene solución?

La respuesta está asociada con las condiciones de frontera dada en la parte (e) de la definición de spline cúbico.

Esto se resume en los 2 teoremas siguientes.

Aproximación de Funciones

Interpolación polinómica a trozos

Teorema: Sea f una función definida en $[a, b]$, entonces f tiene un único spline cúbico S con condición de frontera amarrada, o sea un único spline cúbico que satisface las condiciones $S'(a) = f'(a)$ y $S'(b) = f'(b)$.

Usando $f'(a) = S'_0(a) = S'_0(x_0) = b_0$, la ecuación (39) con $j=0$ se transforma

$$\begin{aligned} f'(a) = b_0 &= \frac{1}{h_0} (a_1 - a_0) - \frac{h_0}{3} (2c_0 + c_1) \\ \Rightarrow 2h_0c_0 + h_0c_1 &= \frac{3}{h_0} (a_1 - a_0) - 3f'(a) \end{aligned} \quad (42)$$

Usando $f'(b) = b_n = S'(x_n)$ y la ecuación (38), se tiene

$$f'(b) = b_n = b_{n-1} + h_{n-1}(c_{n-1} + c_n),$$

La ecuación (39) con $j=n-1$ se transforma

$$b_{n-1} = \frac{1}{h_{n-1}} (a_n - a_{n-1}) - \frac{h_{n-1}}{3} (2c_{n-1} + c_n)$$

Aproximación de Funciones

Interpolación polinómica a trozos

Combinando estas 2 últimas ecuaciones, se obtiene

$$\begin{aligned}
 f'(b) &= \frac{1}{h_{n-1}}(a_n - a_{n-1}) - \frac{h_{n-1}}{3}(2c_{n-1} + c_n) + h_{n-1}(c_{n-1} + c_n) \\
 &= \frac{1}{h_{n-1}}(a_n - a_{n-1}) - \frac{h_{n-1}}{3}(c_{n-1} + 2c_n) \\
 \Rightarrow h_{n-1}c_{n-1} + 2h_{n-1}c_n &= 3f'(b) - \frac{3}{h_{n-1}}(a_n - a_{n-1})
 \end{aligned} \tag{43}$$

El sistema lineal (41) en conjunto con (42) y (43) conduce a un sistema lineal $Ax = b$ donde A es una matriz $(n+1) \times (n+1)$ diagonal dominante estricta, y x como en (41) y b similar.

A es no singular

Aproximación de Funciones

Interpolación polinómica a trozos

Ejemplo del archivo “spline_ejemplo.m”

SPLINE Cubic spline data interpolation

`YY = SPLINE(X,Y,XX)` uses cubic spline interpolation to find `YY`, the values of the underlying function `Y` at the points in the vector `XX`. The vector `X` specifies the points at which the data `Y` is given.

`PP = SPLINE(X,Y)` returns the piecewise polynomial form of the cubic spline interpolant for later use with `PPVAL` and the spline utility `UNMKPP`.

Ordinarily, the not-a-knot end conditions are used. However, if `Y` contains two more values than `X` has entries, then the first and last value in `Y` are used as the endslopes for the cubic spline.

PPVAL Evaluate piecewise polynomial:

`V = PPVAL(PP,XX)` returns the value at the points `XX` of the piecewise polynomial contained in `PP`, as constructed by `SPLINE` or the spline utility `MKPP`.

```
x = -4:4; y = [0 .15 1.12 2.36 2.36 1.46 .49 .06 0];
cs = spline(x,[0 y 0]);
xx = linspace(-4,4,101);
plot(x,y,'o',xx,ppval(cs,xx),'-');
```

```
Estructura cs:
form: 'pp'
breaks: [-4 -3 -2 -1 0 1 2 3 4]
coefs: [8x4 double]
pieces: 8
order: 4
dim: 1
```

Aproximación de Funciones

Polinomio interpolante de Hermite

Ejemplo. Se plantea el problema de encontrar un polinomio p que cumpla las condiciones $p(0) = 0$, $p(1) = 1$, $p'(1/2) = 2$

Dado que se imponen 3 condiciones es razonable pensar que el polinomio buscado es de grado 2, digamos

$$p(x) = a + bx + cx^2$$

Las condiciones dadas implican que

$$0 = p(0) = a$$

$$1 = p(1) = a + b + c = b + c$$

$$2 = p'(1/2) = b + c$$

} contradicción

Por lo tanto no existe un polinomio de grado 2 que cumpla las condiciones dadas. Si se considera un polinomio de grado 3 que cumpla las 3 condiciones dadas, es fácil verificar que existen infinitas soluciones del problema.

Aproximación de Funciones

Polinomio interpolante de Hermite

Definición. Dados $n+1$ números distintos x_0, x_1, \dots, x_n y los enteros no negativos m_0, m_1, \dots, m_n , un **polinomio osculante o de Hermite** que aproxima a una función $f \in C^m[a,b]$, donde $m = \max[m_0, m_1, \dots, m_n]$ y $x_i \in [a,b]$ para cada $i=0, \dots, n$, es el polinomio de menor grado con la propiedad de que coincide con la función f y todas sus derivadas de orden menor o igual a m_i en x_i para cada $i = 0, 1, \dots, n$.

El grado de este polinomio será a lo más

$$M = \sum_{i=0}^n m_i + n,$$

ya que el número de condiciones a satisfacer es $\sum_{i=0}^n m_i + (n + 1)$,

y un polinomio de grado M tiene $M+1$ coeficientes que pueden usarse para satisfacer estas condiciones.

Obs. Existe un único polinomio que cumple las condiciones dadas en la definición anterior.

Aproximación de Funciones

Polinomio interpolante de Hermite

Obs.

- Cuando $n=0$, el polinomio osculante que aproxima a f es simplemente el polinomio de Taylor de grado m_0 .
- Cuando $m_i=0$ para $i=0, \dots, n$, el polinomio osculante es el polinomio que interpola a f en x_0, x_1, \dots, x_n , es decir el polinomio de interpolación.
- El conjunto de los polinomios osculantes es una generalización de los polinomios de interpolación.
- Cuando $m_i=1$ para cada $i=0, \dots, n$, da una clase de polinomios particulares de Hermite. Para una función f , estos polinomios no sólo coinciden con f en x_0, x_1, \dots, x_n , sino que, como sus primeras derivadas coinciden también con las de f , tienen la misma apariencia que la función f en $(x_i, f(x_i))$ (las rectas tangentes al polinomio y a la función coinciden).

Aproximación de Funciones

Polinomio interpolante de Hermite

Teorema. Si $f \in C^m[a,b]$ y $x_0, x_1, \dots, x_n \in [a,b]$ son distintos, el único polinomio de menor grado que coincide con f y f' en x_0, x_1, \dots, x_n es un polinomio de grado a lo mas $2n+1$ dado por

$$H_{2n+1}(x) = \sum_{j=0}^n f(x_j) H_{n,j}(x) + \sum_{j=0}^n f'(x_j) \hat{H}_{n,j}(x),$$

donde $H_{n,j}(x) = [1 - 2(x - x_j)L'_{n,j}(x_j)]L_{n,j}^2(x)$ y $\hat{H}_{n,j}(x) = (x - x_j)L_{n,j}^2(x)$.

Aquí, $L_{n,j}$ es el j -ésimo coeficiente polinomial de Lagrange de grado n

$$L_j(x) = \prod_{\substack{i=0 \\ i \neq j}}^{i=n} \frac{x - x_i}{x_j - x_i}.$$

Además, si $f \in C^{2n+2}[a,b]$ entonces

$$f(x) - H_{2n+1}(x) = \frac{(x - x_0)^2 \cdots (x - x_n)^2}{(2n + 2)!} f^{(2n+2)}(\xi)$$

para algún ξ con $a < \xi < b$.

Aproximación de Funciones

Polinomio interpolante de Hermite

Obs. Aún cuando el teorema anterior da una descripción completa de los polinomios de Hermite, determinar y evaluar los polinomios de Lagrange y sus derivadas resulta un procedimiento tedioso, aún para valores pequeños de n .

Un método alternativo para generar aproximaciones de Hermite está basado en la diferencia dividida interpolante de Newton para el polinomio de Lagrange en x_0, x_1, \dots, x_n .

$$p(x) = f[x_0] + \sum_{k=1}^n f[x_0, \dots, x_k] \prod_{j=0}^{k-1} (x - x_j),$$

y la conexión entre la enésima diferencia dividida y la enésima derivada de f , es decir,

$$f[x_0, \dots, x_n] = \frac{f^{(n)}(\xi)}{n!}.$$

Aproximación de Funciones

Polinomio interpolante de Hermite

Supongamos que se dan $n+1$ puntos distintos x_0, x_1, \dots, x_n , junto con sus valores de f y f' . Definimos una nueva sucesión $z_0, z_1, \dots, z_{2n+1}$ por

$$z_{2i} = z_{2i+1} = x_i, \quad \text{para } i = 0, \dots, n.$$

nodo	diferencia dividida				
	cero	uno	dos	tres	...
$z_0 = x_0$	$f[z_0] = f(x_0)$				
$z_1 = x_0$	$f[z_1] = f(x_0)$	$f[z_0, z_1] = f'(x_0)$			
$z_2 = x_1$	$f[z_2] = f(x_1)$	$f[z_1, z_2]$	$f[z_0, z_1, z_2]$		
$z_3 = x_1$	$f[z_3] = f(x_1)$	$f[z_2, z_3] = f'(x_1)$	$f[z_1, z_2, z_3]$	$f[z_0, z_1, z_2, z_3]$	
$z_4 = x_2$	$f[z_4] = f(x_2)$	$f[z_3, z_4]$	$f[z_2, z_3, z_4]$	$f[z_1, z_2, z_3, z_4]$...
$z_5 = x_2$	$f[z_5] = f(x_2)$	$f[z_4, z_5] = f'(x_2)$	$f[z_3, z_4, z_5]$	$f[z_2, z_3, z_4, z_5]$	

$$H_{2n+1}(x) = f[z_0] + \sum_{k=1}^{2n+1} f[z_0, \dots, z_k] \prod_{j=0}^{k-1} (x - z_j),$$

Aproximación de Funciones

Polinomio interpolante de Hermite

Determinar el polinomio de Hermite que toma los valores:

$$p(1) = 2, \quad p'(1) = 3, \quad p(2) = 6, \quad p'(2) = 7, \quad p''(2) = 4$$

$$p(x) = c_0 + c_1(x - 1) + c_2(x - 1)^2 + c_3(x - 1)^2(x - 2) + c_4(x - 1)^2(x - 2)^2$$

Aproximación de Funciones

$$f[x_i, \dots, x_{i+k}] = \frac{f[x_{i+1}, \dots, x_{i+k}] - f[x_i, \dots, x_{i+k-1}]}{x_{i+k} - x_i}$$

$$f[x_0, \dots, x_n] = \frac{f^{(n)}(\xi)}{n!}$$

Polinomio interpolante de Hermite

Determinar el polinomio de Hermite que toma los valores:

$p(1) = 2, \quad p'(1) = 3, \quad p(2) = 6, \quad p'(2) = 7, \quad p''(2) = 4$

diferencia dividida

nodo	cero	uno	dos	tres	cuatro
$z_0 = 1$	$f[z_0] = 2$				
		$f[z_0, z_1] = f'(z_0) = 3$			
			$f[z_0, z_1, z_2] = \frac{4-3}{2-1} = 1$		
$z_1 = 1$	$f[z_1] = 2$				
		$f[z_1, z_2] = \frac{6-2}{2-1} = 4$			
			$f[z_1, z_2, z_3] = \frac{7-4}{2-1} = 3$		
				$f[z_0, z_1, z_2, z_3] = 2$	
$z_2 = 2$	$f[z_2] = 6$				
		$f[z_2, z_3] = f'(z_2) = 7$			
			$f[z_2, z_3, z_4] = \frac{1}{2} f''(z_2) = 4$		
				$f[z_1, z_2, z_3, z_4] = 1$	
$z_3 = 2$	$f[z_3] = 6$				
		$f[z_3, z_4] = f'(z_3) = 7$			
$z_4 = 2$	$f[z_4] = 6$				

$$p(x) = 2 + 3(x - 1) + (x - 1)^2 + 2(x - 1)^2(x - 2) - (x - 1)^2(x - 2)^2$$

Aproximación de Funciones

Curvas paramétricas

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$$

$$\begin{cases} x(t) = a \cos t \\ y(t) = b \sin t \end{cases}$$

Representación cartesiana

Representación paramétrica

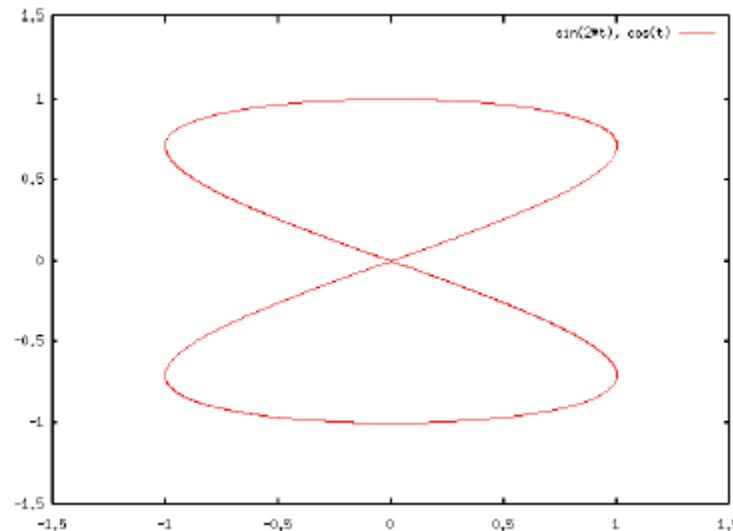
$$r : \mathbb{R} \rightarrow \mathbb{R}^2$$

$$r(t) = (x(t), y(t))$$

Función vectorial
de 1 variable

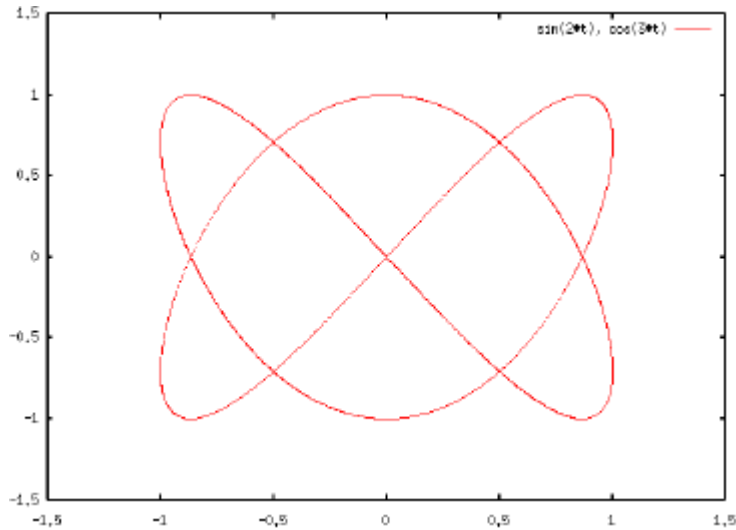
$$\begin{cases} x(t) = \sin(2t) \\ y(t) = \cos t \end{cases}$$

curva de Lissajous



Aproximación de Funciones

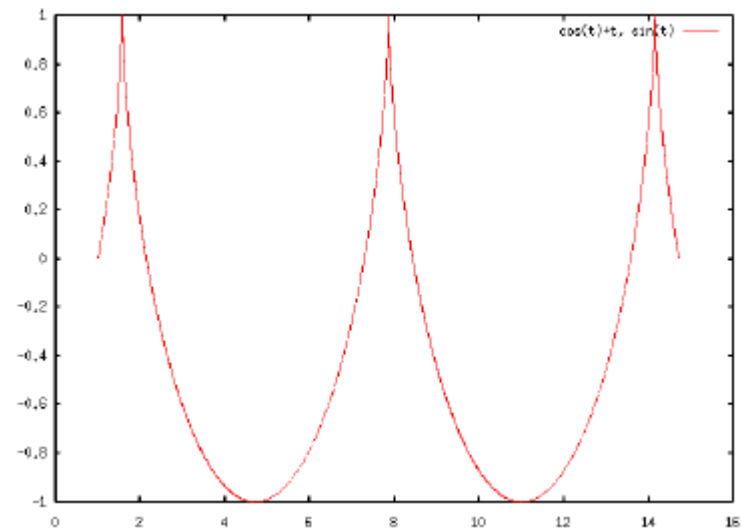
Curvas paramétricas



$$\begin{cases} x(t) = \sin(2t) \\ y(t) = \cos(3t) \end{cases}$$

Curva suave

Curva no suave, pero sus componentes si lo son



$$\begin{cases} x(t) = \cos(t) + t \\ y(t) = \sin t \end{cases}$$

Aproximación de Funciones

Curvas paramétricas

* Lemniscata de Bernoulli (símbolo del *infinito* en matemáticas):

$$x(t) = a \sin(t)/(1 + \cos^2(t)) \quad ; \quad y(t) = a \sin(t)\cos(t)/(1 + \cos^2(t))$$

* Deltoide (Euler)

$$x(t) = a (2 \cos(t) + \cos(2t)) \quad ; \quad y(t) = a (2 \sin(t) - \sin(2t))$$

* Cardiode

$$x(t) = a (2 \cos(t) + \cos(2t)) \quad ; \quad y(t) = a (2 \sin(t) + \sin(2t))$$

única
diferencia

* Astroide

$$x(t) = a \cos^3(t) \quad ; \quad y(t) = a \sin^3(t)$$

Aproximación de Funciones

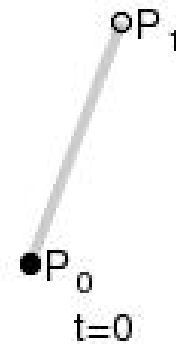
Curvas de Bézier

Se denomina curvas de Bézier a un sistema que se desarrolló hacia los años 1960, para el trazado de dibujos técnicos, en el diseño aeronáutico y de automóviles. Su denominación es en honor a Pierre Bézier, quien ideó un método de descripción matemática de las curvas que se comenzó a utilizar con éxito en los programas de CAD.

Dados los puntos P_0 y P_1 , una **curva lineal de Bézier** es una línea recta entre los dos puntos. La curva viene dada por la expresión:

$$B(t) = P_0 + (P_1 - P_0)t = (1-t)P_0 + tP_1, \quad t \in [0,1]$$

La t en la función para la curva lineal de Bézier se puede considerar como un descriptor de cuán lejos está $B(t)$ de P_0 a P_1 . Por ejemplo cuando $t=0.25$, $B(t)$ es un cuarto de la longitud entre el punto P_0 y el punto P_1 . Como t varía entre 0 y 1, $B(t)$ describe un línea recta de P_0 a P_1 .



Aproximación de Funciones

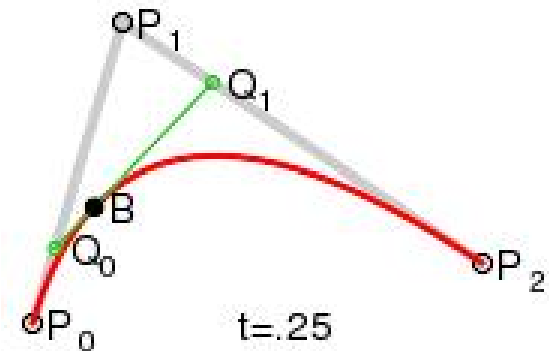
Curvas de Bézier

Una **curva cuadrática de Bézier** es el camino trazado por la función $B(t)$, dados los puntos: P_0 , P_1 y P_2 .

$$B(t) = (1-t)^2 P_0 + 2t(1-t)P_1 + t^2 P_2, \quad t \in [0,1]$$

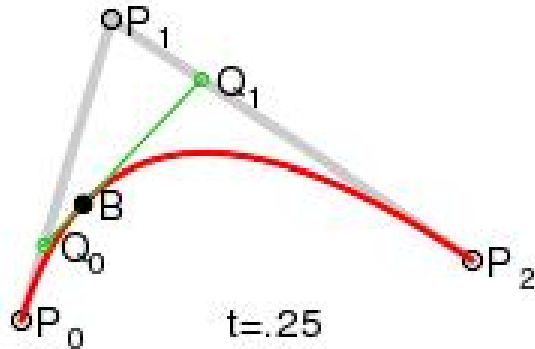
Para curvas cuadráticas se construyen puntos intermedios desde Q_0 a Q_1 tales que t varía de 0 a 1, y que

- Punto Q_0 varía de P_0 a P_1 y describe una curva lineal de Bézier.
- Punto Q_1 varía de P_1 a P_2 y describe una curva lineal de Bézier.
- Punto $B(t)$ varía de Q_0 a Q_1 y describe una curva cuadrática de Bézier.



Aproximación de Funciones

Curvas de Bézier



$$Q_0 = (1-t)P_0 + tP_1, \quad t \in [0,1]$$

$$Q_1 = (1-t)P_1 + tP_2, \quad t \in [0,1]$$

$$B = (1-t)Q_0 + tQ_1, \quad t \in [0,1]$$

$$B(t) = (1-t)\{(1-t)P_0 + tP_1\} + t\{(1-t)P_1 + tP_2\}, \quad t \in [0,1]$$

$$= (1-t)^2 P_0 + t(1-t)P_1 + t(1-t)P_1 + t^2 P_2, \quad t \in [0,1]$$

$$= (1-t)^2 P_0 + 2t(1-t)P_1 + t^2 P_2, \quad t \in [0,1]$$

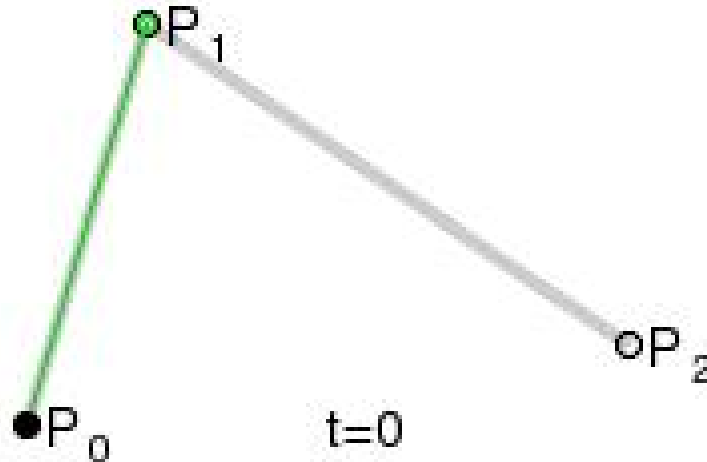
$$B(t) = \sum_{i=0}^2 \binom{2}{i} t^i (1-t)^{2-i} P_i \quad t \in [0,1].$$

Dados los puntos: P_0 , P_1 y P_2 , la curva cuadrática de Bézier es el camino trazado por la función $B(t)$.

Aproximación de Funciones

Curvas de Bézier

Como se genera geoméricamente la **curva cuadrática de Bézier**.



Curva cuadrática de Bézier

$$B(t) = (1-t)^2 P_0 + 2t(1-t) P_1 + t^2 P_2, \quad t \in [0,1]$$

Aproximación de Funciones

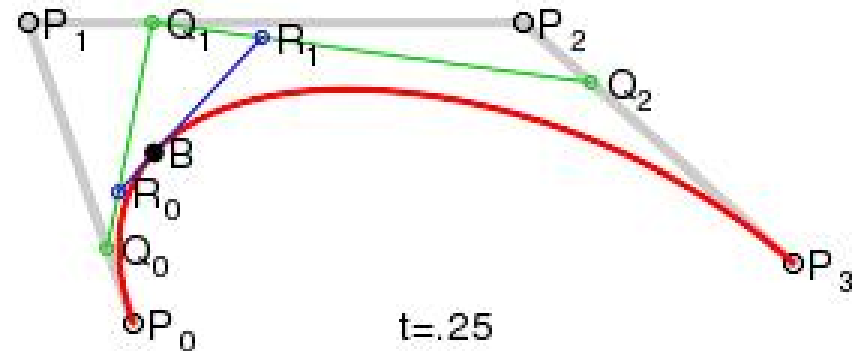
Curvas de Bézier

Una **curva cúbica de Bézier** es el camino trazado por la función $B(t)$, dados los puntos: P_0 , P_1 , P_2 y P_3 .

$$B(t) = (1-t)^3 P_0 + 3t(1-t)^2 P_1 + 3t^2(1-t) P_2 + t^3 P_3, \quad t \in [0,1]$$

$$B(t) = \sum_{i=0}^3 \binom{3}{i} t^i (1-t)^{3-i} P_i \quad t \in [0,1].$$

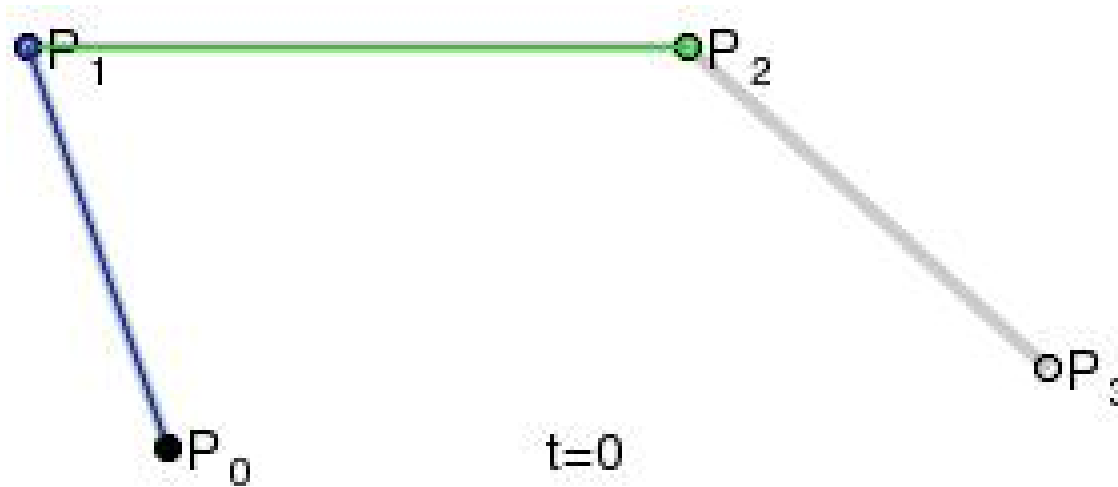
Cuatro puntos del plano o del espacio tridimensional, P_0 , P_1 , P_2 y P_3 definen una curva cúbica de Bézier. La curva comienza en el punto P_0 y se dirige hacia P_1 y llega a P_3 viniendo de la dirección del punto P_2 . Usualmente, no pasará ni por P_1 ni por P_2 . Estos puntos sólo están ahí para proporcionar información direccional. La distancia entre P_0 y P_1 determina "qué longitud" tiene la curva cuando se mueve hacia la dirección de P_2 antes de dirigirse hacia P_3 .



Aproximación de Funciones

Curvas de Bézier

Como se genera geoméricamente la **curva cúbica de Bézier**.



Curva cúbica de Bézier

$$B(t) = (1-t)^3 P_0 + 3t(1-t)^2 P_1 + 3t^2(1-t) P_2 + t^3 P_3, \quad t \in [0,1]$$

Aproximación de Funciones

Curvas de Bézier

La **curva de Bézier de grado n** puede ser generalizada de la siguiente manera. Dados los puntos P_0, P_1, \dots, P_n , la curva de Bézier es del tipo:

$$B(t) = \sum_{i=0}^n \binom{n}{i} t^i (1-t)^{n-i} P_i$$

$$= \binom{n}{0} (1-t)^n P_0 + \binom{n}{1} t(1-t)^{n-1} P_1 + \dots + \binom{n}{n} t^n P_n, \quad t \in [0,1].$$

Esta ecuación puede ser expresada de manera recursiva como sigue: sea la expresión $B_{P_0 \dots P_n}$ que denota la curva de Bézier determinada por los puntos P_0, \dots, P_n . Entonces

$$B(t) = B_{P_0 \dots P_n}(t) = (1-t)B_{P_0 \dots P_{n-1}}(t) + tB_{P_1 \dots P_n}(t), \quad t \in [0,1].$$

En otras palabras, la curva de Bézier de grado n , es una interpolación entre curvas de Bézier de grado $n-1$.

Aproximación de Funciones

Curvas de Bézier

Existe otra terminología asociada a las curvas de Bézier

$$B(t) = \sum_{i=0}^n P_i b_{i,n}(t), \quad t \in [0,1].$$

donde los polinomios

$$b_{i,n}(t) = \binom{n}{i} t^i (1-t)^{n-i}, \quad i = 0, \dots, n,$$

se conocen como los **polinomios de Bernstein** de grado n .

Los puntos P_i son llamados puntos de control para de las curvas de Bézier.

El polígono formado por la conexión de los puntos de Bézier con rectas, comenzando por P_0 y terminando en P_n , se denomina polígono de Bézier (o polígono de control). El “capsula” convexa del polígono de Bézier contiene las curvas de Bézier.

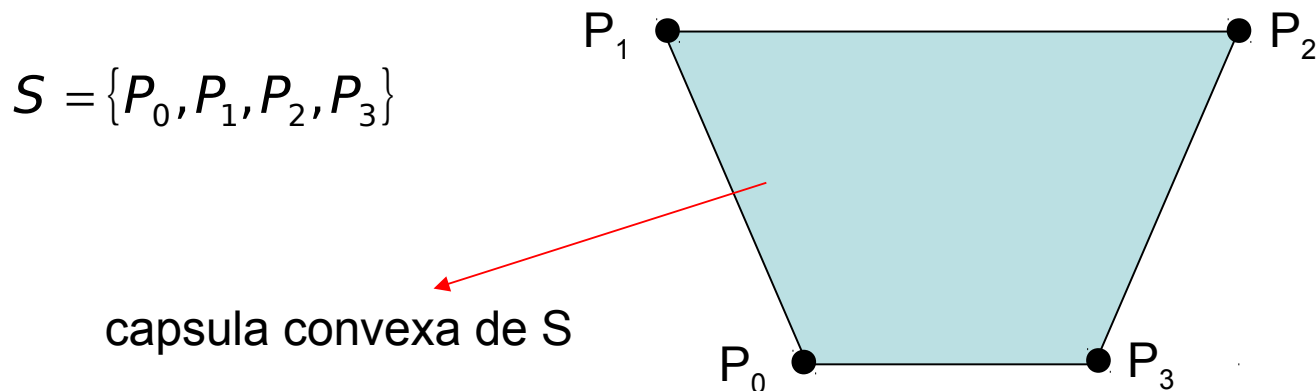
Aproximación de Funciones

Curvas de Bézier

Definición 1. Un conjunto C de puntos es convexo si para todo $p, q \in C$ el segmento de recta está contenido en C .

Definición 2. La cápsula o envolvente convexa de un conjunto S de puntos es el conjunto convexo más pequeño C que contiene a S .

Obs. La cápsula convexa de un conjunto finito de puntos S es un polígono convexo cuyos vértices (puntos de esquina) son elementos de S .

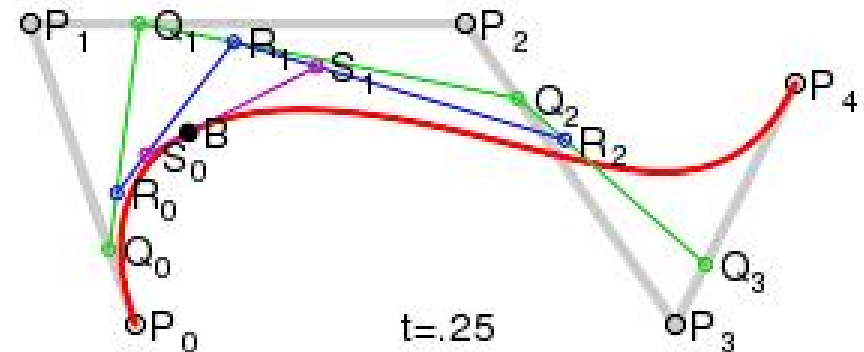


Aproximación de Funciones

Curvas de Bézier

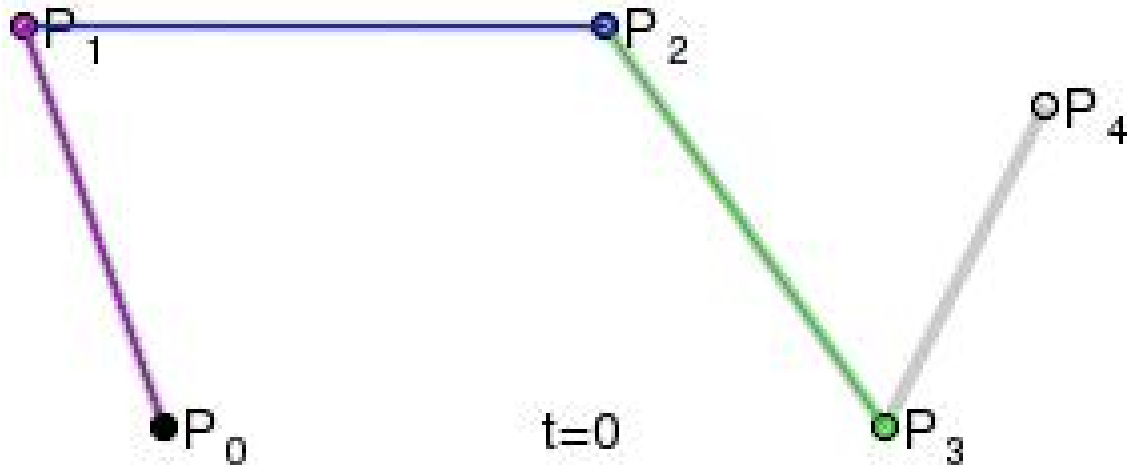
Propiedades de las curvas de Bézier:

- La curva de Bézier se encuentra en el interior de la envolvente convexa de los puntos de control.
- La curva de Bézier es infinitamente derivable.
- La curva comienza en el punto P_0 y termina en el P_n . Esta peculiaridad es llamada “interpolación del punto final”.
- La curva es un segmento recto si, y sólo si, todos los puntos de control están alineados.
- El comienzo (final) de la curva es tangente a la primera (final) sección del polígono de Bézier.



Aproximación de Funciones

Curvas de Bézier



Curva de Bézier de grado 4

$$B(t) = \sum_{i=0}^4 \binom{4}{i} t^i (1-t)^{4-i} P_i \quad t \in [0,1].$$

Aproximación de Funciones

Curvas de Bézier

Forma matricial de Bézier cúbica: matriz y vector de geometría

$$B(t) = (1-t)^3 P_0 + 3t(1-t)^2 P_1 + 3t^2(1-t) P_2 + t^3 P_3, \quad t \in [0,1]$$

$$B(t) = \begin{pmatrix} t^3 & t^2 & t & 1 \end{pmatrix} \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{pmatrix} \quad t \in [0,1].$$

$$\begin{pmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{pmatrix} = \begin{pmatrix} x_0 & y_0 \\ x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \end{pmatrix}$$

CO3211 – Cálculo Numérico

FIN