

# Métodos numéricos para matrices

## Curso de Física Computacional

M. en C. Gustavo Contreras Mayén

Facultad de Ciencias - UNAM

16 de noviembre de 2017



1. Introducción
2. Sistemas lineales
3. Sistemas algebraicos
4. Métodos de solución
5. Tres métodos directos
6. Método de eliminación de Gauss

# 1. Introducción

## 1.1 Primer ejemplo

## 1.2 Segundo ejemplo

# 2. Sistemas lineales

# 3. Sistemas algebraicos

# 4. Métodos de solución

# 5. Tres métodos directos

# 6. Método de eliminación de Gauss

# Primer ejemplo

Supongamos que queremos estudiar el espectro de vibraciones de una molécula con  $n$  grados de libertad.

La primera aproximación consiste en investigar las oscilaciones armónicas del sistema, expandiendo la energía potencial hasta el segundo orden en las coordenadas generalizadas alrededor de las posiciones de equilibrio:

# Primer ejemplo

Supongamos que queremos estudiar el espectro de vibraciones de una molécula con  $n$  grados de libertad.

La primera aproximación consiste en investigar las oscilaciones armónicas del sistema, expandiendo la energía potencial hasta el segundo orden en las coordenadas generalizadas alrededor de las posiciones de equilibrio:

$$U(q_1, q_2, \dots, q_n) \simeq \frac{1}{2} \sum_{i,j=1} A_{ij} q_i q_j$$

# Energía potencial

$$U(q_1, q_2, \dots, q_n) \simeq \frac{1}{2} \sum_{i,j=1} A_{ij} q_i q_j$$

donde  $q_i$  son las coordenadas generalizadas y  $A_{jk}$  son parámetros del potencial que usualmente pueden obtenerse a partir de un cálculo de química cuántica.

# Energía cinética

La energía cinética puede escribirse en términos de las velocidades generalizadas.

$$T(\dot{q}_1, \dot{q}_2, \dots, \dot{q}_n) \simeq \frac{1}{2} \sum_{i,j=1} M_{ij} \dot{q}_i \dot{q}_j$$

donde las  $\dot{q}_i = dq_i/dt$  son las velocidades generalizadas, las  $M_{ij}$  son los elementos de masa generalizado en la matriz, cuyos valores dependen en particular de la molécula.

Aplicando la ecuación de Lagrange

$$\frac{\partial \mathcal{L}}{\partial q_i} - \frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{q}_i} = 0$$

donde el lagrangiano del sistema es:  $\mathcal{L} = T - U$ ,  
por tanto, tenemos que:

$$\sum_{j=1}^n (A_{ij} \dot{q}_j + M_{ij} \ddot{q}_j) = 0, \quad i = 1, 2, \dots, n$$



# Dependencia del tiempo de tipo oscilatorio

Si suponemos que la dependencia del tiempo en las coordenadas generalizadas es de tipo oscilatorio, con

$$q_j = x_j e^{-i\omega t}$$

Se obtiene el siguiente sistema de ecuaciones homogéneo:

$$\sum_{j=1}^n (A_{ij} - \omega^2 M_{ij}) x_j = 0, \quad i = 1, 2, \dots, n$$

# Expresión matricial

La ecuación anterior se puede re-escribir en forma matricial, como.

$$\begin{pmatrix} A_{11} & \dots & A_{1n} \\ \vdots & \vdots & \vdots \\ A_{n1} & \dots & A_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = \lambda \begin{pmatrix} M_{11} & \dots & M_{1n} \\ \vdots & \vdots & \vdots \\ M_{n1} & \dots & M_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$$

De manera equivalente

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{M}\mathbf{x}$$

donde  $\lambda = \omega^2$  es el valor propio y  $\mathbf{x}$  es el correspondiente vector propio de la ecuación de valores propios.

# Solución no trivial del sistema

Este es un sistema de ecuaciones lineales homogéneas; con el fin de contar con una solución no trivial del conjunto de ecuaciones, el determinante de la matriz de coeficientes debe anularse, esto es

$$\det|\mathbf{A} - \lambda\mathbf{M}| = 0$$

# Raíces de la ecuación matricial

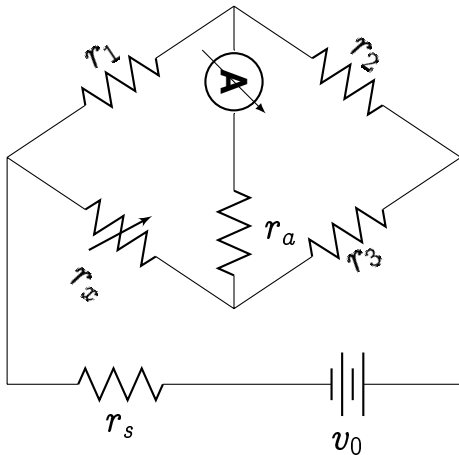
$$\det|\mathbf{A} - \lambda\mathbf{M}| = 0$$

Las raíces de esta ecuación  $\lambda_k$  con  $k = 1, 2, \dots, n$  proporcionan todas las frecuencias angulares vibracionales de la molécula

$$\omega_k = \sqrt{\lambda_k}$$

## Segundo ejemplo

Consideremos el siguiente circuito eléctrico:

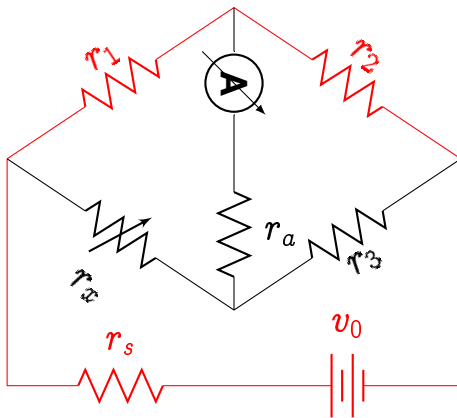


# Solución para el circuito

Podemos aplicar las leyes de Kirchhoff para obtener un conjunto de ecuaciones que relacionan los voltajes y las corrientes del circuito, luego entonces, resolverlo y encontrar las incógnitas.

Pero ahora veamos el caso del puente desbalanceado de Wheastone: tenemos tres circuitos independientes:

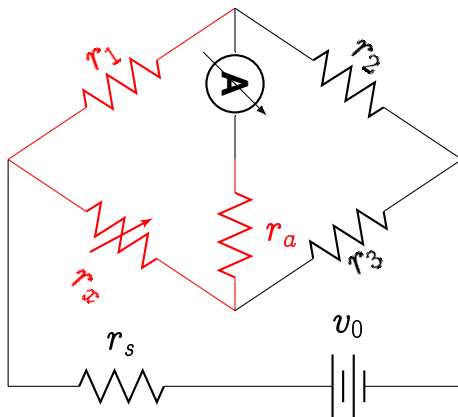
# Circuitos independientes 1



$$r_s i_1 + r_1 i_2 + r_2 i_3 = v_0$$

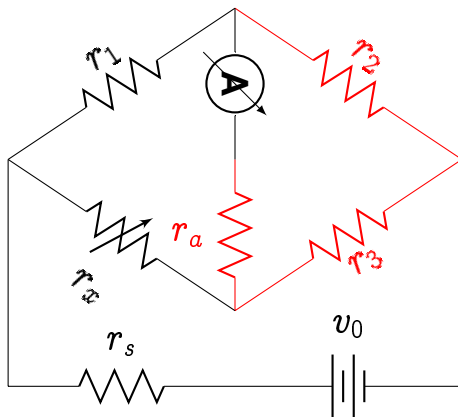


# Circuitos independientes 2



$$-r_x i_1 + (r_1 + r_x + r_a) i_2 + r_a i_3 = 0$$

# Circuitos independientes 3



$$-r_3 i_1 - r_a i_2 + (r_2 + r_3 + r_a) i_3 = 0$$

Entonces el sistema de ecuaciones resulta ser:

$$\begin{aligned}r_s i_1 + r_1 i_2 + r_2 i_3 &= v_0 \\ -r_x i_1 + (r_1 + r_x + r_a) i_2 + r_a i_3 &= 0 \\ -r_3 i_1 - r_a i_2 + (r_2 + r_3 + r_a) i_3 &= 0\end{aligned}$$

# Sistema de ecuaciones

Entonces el sistema de ecuaciones resulta ser:

$$\begin{aligned}r_s i_1 + r_1 i_2 + r_2 i_3 &= v_0 \\ -r_x i_1 + (r_1 + r_x + r_a) i_2 + r_a i_3 &= 0 \\ -r_3 i_1 - r_a i_2 + (r_2 + r_3 + r_a) i_3 &= 0\end{aligned}$$

El sistema anterior se puede escribir como

$$\mathbf{R}\mathbf{i} = \mathbf{v}$$

# Descripción del sistema

$$\mathbf{R}\mathbf{i} = \mathbf{v}$$

donde la matriz de coeficientes de las resistencias es

$$\mathbf{R} = \begin{pmatrix} r_s & r_1 & r_2 \\ -r_x & r_1 + r_x + r_a & -r_a \\ -r_3 & -r_a & r_2 + r_3 + r_a \end{pmatrix}$$

# Descripción del sistema

y los vectores columna de corrientes y voltajes son:

$$\mathbf{i} = \begin{pmatrix} i_1 \\ i_2 \\ i_3 \end{pmatrix} \quad \mathbf{v} = \begin{pmatrix} v_0 \\ 0 \\ 0 \end{pmatrix}$$

# Solución del sistema matricial

Multiplicando en ambos lados por  $\mathbf{R}^{-1}$  (la matriz inversa de  $\mathbf{R}$ ), la ecuación

$$\mathbf{R}\mathbf{i} = \mathbf{v}$$

resulta

$$\mathbf{i} = \mathbf{R}^{-1}\mathbf{v}$$

que requiere que conozcamos  $\mathbf{R}^{-1}$ .

¿Será necesario conocer siempre la matriz inversa?

# 1. Introducción

## 2. Sistemas lineales

2.1 Sistemas lineales con la computadora

2.2 Unicidad de la solución

2.3 Problemas mal condicionados

2.4 Número de condición de la matriz

## 3. Sistemas algebraicos

## 4. Métodos de solución

## 5. Tres métodos directos



# Sistemas lineales

En este tema analizamos la solución de sistemas de ecuaciones lineales algebraicas, con  $n$  incógnitas.

Cuando tomamos un problema físico, los conjuntos de ecuaciones a menudo son muy grandes, por lo que consumen una gran cantidad de recursos computacionales.

Por lo general, se reducen los requerimientos de almacenamiento y el tiempo de ejecución mediante el aprovechamiento de propiedades especiales de la matriz de coeficientes, tales como poca densidad (la mayoría de los elementos de una matriz dispersa son cero -las llamadas matrices *sparse*-)

Existen muchos algoritmos dedicados a la solución de grandes conjuntos de ecuaciones, cada uno de ellos está adaptado a una determinada forma del coeficiente de la matriz (simétrica, de bandas, escaso, etc.)

En internet se encuentran disponibles librerías para resolver cierto tipo de problemas, al usarlas se requiere conocer la manera en que se proporcionan los argumentos y las variables de resultado que proporcionan, se obliga la revisión de la documentación de cada uno de los códigos.

# Definición de sistema algebraico

Un sistema algebraico tiene la forma

$$A_{11}x_1 + A_{12}x_2 + \dots + A_{1n}x_n = b_1$$

$$A_{21}x_1 + A_{22}x_2 + \dots + A_{2n}x_n = b_2$$

$$= \vdots$$

$$A_{n1}x_1 + A_{n2}x_2 + \dots + A_{nn}x_n = b_n$$

o sencillamente

$$Ax = b$$

# Matriz aumentada

Una representación particularmente útil de las ecuaciones para los propósitos computacionales es la matriz de coeficientes aumentada, que se obtiene al juntar el vector constante  $b$  a la matriz de coeficientes  $A$  de la siguiente manera:

$$[A|b] = \left[ \begin{array}{cccc|c} A_{11} & A_{12} & \dots & A_{1n} & b_1 \\ A_{21} & A_{22} & \dots & A_{2n} & b_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ A_{n1} & A_{n2} & \dots & A_{nn} & b_n \end{array} \right]$$

# Unicidad de la solución

Un sistema de  $n$  ecuaciones lineales con  $n$  incógnitas tiene una solución única, siempre que el determinante de la matriz de coeficientes sea **no singular**, es decir,  $|A| \neq 0$ .

Las filas y columnas de una matriz no singular son linealmente independientes en el sentido de que no hay ninguna fila (o columna) sea una combinación lineal de las otras filas (o columnas).

# Soluciones infinitas

Si la matriz de coeficientes es singular, las ecuaciones pueden tener un número infinito de soluciones o no soluciones en absoluto, dependiendo del vector constante.



# Caso 1

Como ejemplo veamos las ecuaciones

$$2x + y = 3$$

$$4x + 2y = 6$$

La segunda ecuación puede obtenerse multiplicando la primera ecuación por dos, cualquier combinación de  $x$  e  $y$  que satisface la primera ecuación es también una solución de la segunda. El número de tales combinaciones es infinito.

# Las dos rectas se superponen

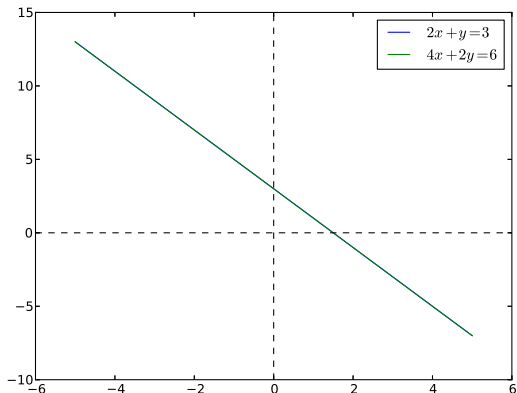


Figura 1: Representación gráfica cuando la matriz de coeficientes es singular.

## Caso 2

En otro caso, las ecuaciones

$$2x + y = 3$$

$$4x + 2y = 0$$

No tienen solución, ya que la ecuación equivalente  $2x + y = 0$ , contradice la primera.

Por tanto, cualquier solución que satisface la primera, no puede satisfacer la segunda.

# Las dos rectas son paralelas

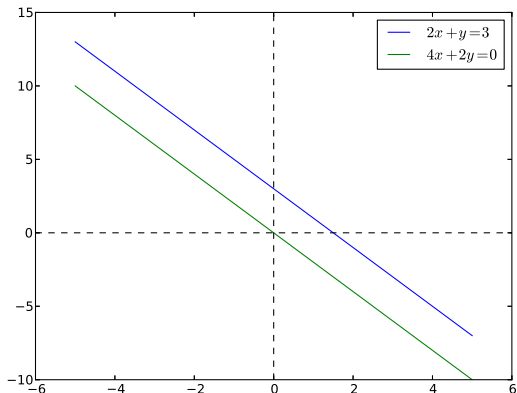


Figura 2: Representación gráfica de un sistema algebraico sin solución.

Una pregunta obvia es: ¿qué sucede cuando la matriz de coeficientes es casi singular; es decir, si  $|A|$  es muy pequeño?

Con el fin de concluir si el determinante de la matriz es pequeño, necesitamos una referencia contra la cual el determinante se pueda medir.

# Norma de una matriz

Esta referencia se denomina la **norma de la matriz** y se denota por  $\| A \|$ . Entonces podemos decir que el determinante es pequeño si

$$|A| \ll \| A \|$$

# Definición de la norma de una matriz

Existen diferentes maneras de calcular la norma de una matriz, tales como

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

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

# Número de condición de la matriz

Una medida formal de condicionamiento es el **Número de condición de la matriz**, definido por:

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

Si este número es cercano a la unidad, la matriz está bien condicionada.



El número de condición aumenta con el grado de mal acondicionamiento, tendiendo a infinito para una matriz singular.

Toma en cuenta que el número de condición no es único, sino que depende de la elección de la norma de la matriz.

Desafortunadamente, el número de condición es costoso de calcular para grandes matrices.

En la mayoría de los casos es suficiente para medir el condicionamiento comparando el determinante con las magnitudes de los elementos en la matriz.

Si las ecuaciones están mal condicionadas, pequeños cambios en la matriz de coeficientes dan como resultado grandes cambios en la solución.

A modo de ejemplo, consideremos las ecuaciones:

$$\begin{aligned}2x + y &= 3 \\ 2x + 1.001y &= 0\end{aligned}$$

que tiene la solución  $x = 1501.5, y = -3000$

# Cuando la norma es menor que los coeficientes

dado que  $|A| = 2(1.001) - 2(1) = 0.002$  es mucho menor que los coeficientes, las ecuaciones están mal condicionadas.

Usemos arreglos y las funciones de `python` para explorar este sistema.

# Usando numpy

```
>>> from numpy import array, linalg  
>>> a = array([[2., 1.], [2, 1.001]])  
>>> b = array([3., 0.]
```

# Usando numpy

```
>>> from numpy import array, linalg  
>>> a = array([[2., 1.], [2, 1.001]])  
>>> b = array([3., 0.]
```

Calculamos el valor del determinante

```
>>> print (linalg.det(a))  
0.002
```

# Seguimos usando el módulo `linalg`

Ahora calculamos el valor del número de condición

```
>>> print (linalg.cond(a))  
5001.00030004
```

# Seguimos usando el módulo `linalg`

Ahora calculamos el valor del número de condición

```
>>> print (linalg.cond(a))  
5001.00030004
```

Y resolvemos el sistema algebraico

```
>>> print (linalg.solve(a,b))  
[ 1501.5 -3000. ]
```



# El sistema está mal condicionado

El efecto de los malos acondicionamientos se puede verificar mediante el cambio de la segunda ecuación

$$2x + y = 3$$

$$2x + 1.002y = 0$$

```
>>> c = array([[2., 1.], [2., 1.002]])
```

# El sistema está mal condicionado

El efecto de los malos acondicionamientos se puede verificar mediante el cambio de la segunda ecuación

$$2x + y = 3$$

$$2x + 1.002y = 0$$

```
>>> c = array([[2., 1.], [2., 1.002]])
```

```
>>> print (linalg.cond(c))
```

```
2501.00060016
```

# El sistema está mal condicionado

El efecto de los malos acondicionamientos se puede verificar mediante el cambio de la segunda ecuación

$$2x + y = 3$$

$$2x + 1.002y = 0$$

```
>>> c = array([[2., 1.], [2., 1.002]])
```

```
>>> print (linalg.cond(c))
```

```
2501.00060016
```

```
>>> print (linalg.solve(c,b))
```

```
[ 751.5 -1500. ]
```

# Un pequeño cambio provoca un enorme impacto

El resultado es  $x = 751.5$ ,  $y = -1500$ .

Nótese que un cambio del 0.1 % en el coeficiente de  $y$  produce un cambio de 100 % en la solución!

1. Introducción

2. Sistemas lineales

3. Sistemas algebraicos

3.1 Definiciones

4. Métodos de solución

5. Tres métodos directos

6. Método de eliminación de Gauss

# Sistemas lineales

Las ecuaciones algebraicas lineales, se presentan en casi todas las ramas del análisis numérico.

Su aplicación más visible es en el análisis de sistemas lineales (cualquier sistema cuya respuesta es proporcional a la entrada se considera que es lineal).

# Sistemas lineales

Los sistemas lineales incluyen estructuras, sólidos, elásticos, flujo de calor, filtraciones de líquidos, campos electromagnéticos, los circuitos eléctricos, etc.

Si el sistema es discreto, tal como una viga o un circuito eléctrico, entonces su análisis conduce directamente a ecuaciones algebraicas lineales.

# Cambio a un sistema discreto

El comportamiento de los sistemas continuos se describen por ecuaciones diferenciales, en lugar de ecuaciones algebraicas.

Sin embargo, como el análisis numérico sólo puede ocuparse de variables discretas, es primero necesario aproximar una ecuación diferencial con un sistema de ecuaciones algebraicas.



# Métodos conocidos

Son conocidos los métodos de diferencias finitas, elementos finitos y métodos de contorno del elemento para el análisis de esta manera.

Estos métodos utilizan diferentes aproximaciones para conseguir una *discretización*, pero en cada caso la tarea final es el mismo: resolver un sistema (a menudo un sistema muy grande) de ecuaciones lineales algebraicas.

# Expresiones de los sistemas algebraicos

El modelado de sistemas lineales invariablemente da lugar a las ecuaciones de la forma  $Ax = b$ , donde  $b$  es la entrada y  $x$  representa la respuesta del sistema.

El coeficiente de la matriz  $A$ , refleja las características del sistema, es independiente de la entrada.

# Expresiones de los sistemas algebraicos

En otras palabras, si se cambia la entrada, las ecuaciones tienen que ser resueltas de nuevo con una  $b$  diferente, pero el mismo  $A$ .

Por lo tanto, es deseable tener un algoritmo que resuelva a ecuación la solución de algoritmo que puede gestionar cualquier número de vectores constantes con un mínimo esfuerzo computacional.

# 1. Introducción

## 2. Sistemas lineales

## 3. Sistemas algebraicos

## 4. Métodos de solución

### 4.1 Solución directa o indirecta

### 4.2 Métodos directos

### 4.3 Métodos indirectos

## 5. Tres métodos directos

Hay dos clases de métodos para resolver sistemas de ecuaciones lineales algebraicas:

- 1 Métodos directos.

Hay dos clases de métodos para resolver sistemas de ecuaciones lineales algebraicas:

- ➊ Métodos directos.
- ➋ Métodos indirectos.

La característica común de los métodos directos es que se transforman las ecuaciones originales en ecuaciones equivalentes (ecuaciones que tienen la misma solución) que se pueden resolver más fácilmente.

# Operaciones elementales

La transformación se lleva a cabo por la aplicación de las tres operaciones citadas a continuación.

Estas *operaciones elementales* no cambian la solución, pero pueden afectar el determinante de la matriz de coeficientes como se indica en el paréntesis.



# Operaciones elementales

- 1 Intercambiar dos ecuaciones (cambia el signo de  $|A|$ )

# Operaciones elementales

- 1 Intercambiar dos ecuaciones (cambia el signo de  $|A|$ )
- 2 Multiplicar una ecuación por una constante distinta de cero (se multiplica  $|A|$  por la misma constante)

# Operaciones elementales

- 1 Intercambiar dos ecuaciones (cambia el signo de  $|A|$ )
- 2 Multiplicar una ecuación por una constante distinta de cero (se multiplica  $|A|$  por la misma constante)
- 3 Multiplicar una ecuación por una constante distinta de cero y luego restarlo de otra ecuación (deja a  $|A|$  sin cambios)

Los métodos iterativos, o métodos indirectos, comienzan con una suposición de la solución  $x$ , y luego repetidamente refinan la solución hasta alcanzar un criterio de convergencia.

Los métodos iterativos son generalmente menos eficientes que sus homólogos directos debido al gran número de iteraciones necesarias.

Pero sí tienen ventajas computacionales importantes si la matriz de coeficientes es muy grande y escasamente pobladas (la mayoría de los coeficientes son cero).

# 1. Introducción

## 2. Sistemas lineales

## 3. Sistemas algebraicos

## 4. Métodos de solución

## 5. Tres métodos directos

### 5.1 Uso de matrices $U$ , $L$ , $I$

### 5.2 Operación con matrices

## 6. Método de eliminación de Gauss

# Tres métodos directos

A continuación se enumeran los tres métodos directos más usados, cada uno de los cuales utiliza operaciones elementales para producir su propia manera de resolver ecuaciones.

# Tres métodos directos

Método	Forma inicial	Forma final
Eliminación de Gauss	$Ax = b$	$Ux = c$
Descomposición LU	$Ax = b$	$LUx = b$
Eliminación Gauss-Jordan	$Ax = b$	$Ix = c$

Donde  $U$  representa una matriz triangular superior,  $L$  es una matriz triangular inferior e  $I$  es la matriz identidad. Una matriz cuadrada se llama triangular si contiene sólo elementos cero en un lado de la diagonal principal.



# Ejemplos de matrices $U$ y $L$

Una matriz triangular superior de  $3 \times 3$  tiene forma

$$U = \begin{bmatrix} U_{11} & U_{12} & U_{13} \\ 0 & U_{22} & U_{23} \\ 0 & 0 & U_{33} \end{bmatrix}$$

y una matriz triangular inferior de  $3 \times 3$ :

$$L = \begin{bmatrix} L_{11} & 0 & 0 \\ L_{21} & L_{22} & 0 \\ L_{31} & L_{32} & L_{33} \end{bmatrix}$$

# Matrices triangulares

Las matrices triangulares juegan un papel importante en el álgebra lineal, ya que simplifican muchos cálculos.

Por ejemplo, consideremos la ecuación  $Lx = c$ :

$$L_{11}x_1 = c_1$$

$$L_{21}x_1 + L_{22}x_2 = c_2$$

$$L_{31}x_1 + L_{32}x_2 + L_{33}x_3 = c_3$$

$$\vdots$$

# Matrices triangulares

Si se resuelven las ecuaciones hacia delante, comenzando con la primera ecuación, los cálculos son muy fáciles, ya que cada ecuación contiene sólo una incógnita a la vez.

# Sustitución hacia adelante

La solución se obtiene haciendo:

$$x_1 = \frac{c_1}{L_{11}}$$

$$x_2 = \frac{(c_2 - L_{21}x_1)}{L_{22}}$$

$$x_3 = \frac{(c_3 - L_{31}x_1 - L_{32}x_2)}{L_{33}}$$

...

Este procedimiento se conoce como *sustitución hacia adelante*.

# Sustitución hacia atrás

De una manera similar la operación  $Ux = c$ , se encuentra en el proceso de eliminación de Gauss, y puede resolverse fácilmente por sustitución hacia atrás, que se inicia con la última ecuación y va retrocediendo a través de las ecuaciones.

# Sustitución hacia atrás

Las ecuaciones  $LUx = b$ , que están asociadas con la descomposición  $LU$ , también pueden resolverse rápidamente si las sustituimos con dos conjuntos de ecuaciones equivalentes:

$$Ly = b \text{ y } Ux = y.$$

# Sustitución hacia atrás

Ahora  $Ly = b$  se puede resolverse para  $y$  por sustitución hacia delante, seguido de la solución de  $Ux = y$  por medio de la sustitución hacia atrás.

# Operaciones con la matriz identidad

Las ecuaciones  $Ix = c$ , que se generan en la eliminación Gauss-Jordan, son equivalentes a  $x = c$  (recordemos la identidad  $Ix = x$ ), de modo que  $c$  es la solución.



# Ejercicio 1

Determinar si la siguiente matriz es singular:

$$A = \begin{bmatrix} 2.1 & -0.6 & 1.1 \\ 3.2 & 4.7 & -0.8 \\ 3.1 & -6.5 & 4.1 \end{bmatrix}$$

# Solución al ejercicio 1

Usando lo que hemos visto respecto al módulo `numpy.linalg`.

```
>>> A = array([[2.1, -0.6, 1.1], \
[3.2, 4.7, -0.8], \
[3.1, -6.5, 4.1]])
```

# Solución al ejercicio 1

Usando lo que hemos visto respecto al módulo `numpy.linalg`.

```
>>> A = array([[2.1, -0.6, 1.1], \
               [3.2, 4.7, -0.8], \
               [3.1, -6.5, 4.1]])
```

```
>>> A
array([[ 2.1, -0.6,  1.1],
       [ 3.2,  4.7, -0.8],
       [ 3.1, -6.5,  4.1]])
```

# Solución al ejercicio 1

Usando lo que hemos visto respecto al módulo `numpy.linalg`.

```
>>> A = array([[2.1, -0.6, 1.1], \
[3.2, 4.7, -0.8], \
[3.1, -6.5, 4.1]])
```

```
>>> A
array([[ 2.1, -0.6,  1.1],
       [ 3.2,  4.7, -0.8],
       [ 3.1, -6.5,  4.1]])
```

```
>>> print (linalg.det(A))
0.0
```

# Resultado

$$A = \begin{bmatrix} 2.1 & -0.6 & 1.1 \\ 3.2 & 4.7 & -0.8 \\ 3.1 & -6.5 & 4.1 \end{bmatrix}$$

La matriz  $A$  es singular, ya que el determinante vale cero.

## Ejercicio 2

Resolver la ecuación  $Ax = b$ , donde

$$A = \begin{bmatrix} 8 & -6 & 2 \\ -4 & 11 & -7 \\ 4 & -7 & 6 \end{bmatrix} \quad b = \begin{bmatrix} 28 \\ -40 \\ 33 \end{bmatrix}$$

Obtenemos ahora con la librería `numpy.linalg`, las matrices  $L$  y  $U$ :

Código 1: Código para obtener las matrices LU

```
1 from numpy import array, linalg
2
3 A = array([[8., -6., 2.], [-4., 11.,
4            -7.], [4., -7, 6.]])
5
6 P, L, U = linalg.lu(A)
7
8 print ('P')
9 print (P)
10
11 print ('L')
12 print (L)
```

```
12 |  
13 | print ('U')  
14 | print (U)
```



Sabiendo que la descomposición  $LU$  de la matriz de coeficientes es:

$$A = LU = \begin{bmatrix} 1. & 0. & 0. \\ -0.5 & 1. & 0. \\ 0.5 & -0.5 & 1 \end{bmatrix} \begin{bmatrix} 8. & -6. & 2. \\ 0. & 8. & 6. \\ 0. & 0. & 2. \end{bmatrix}$$

# Recuperando la matriz inicial $A$

Para verificar que efectivamente el producto de las matrices  $L$  y  $U$  nos regresa la matriz  $A$ , basta con que realicemos el producto de matrices, pero cuidado, si hacemos:

```
>>> print (P*U)
array([[ 8., -0.,  0.],
       [ 0.,  8., -0.],
       [ 0.,  0.,  2.]])
```

# Una operación que no devuelve lo esperado

Lo que obtenemos es una matriz que es el producto “entrada” por “entrada” de las matrices  $L$  y  $U$ , por lo que no es nuestra matriz inicial  $A$ .

# Una operación que no devuelve lo esperado

Lo que obtenemos es una matriz que es el producto “entrada” por “entrada” de las matrices  $L$  y  $U$ , por lo que no es nuestra matriz inicial  $A$ .

La manera de recuperar la matriz  $A$ , es mediante el correcto uso del producto de dos arreglos, para esto debemos de utilizar la función `dot`, recordemos que se puede usar hasta de tres maneras distintas:

# Operacion con escalares y matrices

- 1 Cuando multiplicamos un escalar por un escalar, lo que nos devuelve un escalar.

# Operacion con escalares y matrices

- 1 Cuando multiplicamos un escalar por un escalar, lo que nos devuelve un escalar.
- 2 Cuando multiplicamos un escalar por un arreglo, lo que nos devuelve un arreglo.

# Operacion con escalares y matrices

- 1 Cuando multiplicamos un escalar por un escalar, lo que nos devuelve un escalar.
- 2 Cuando multiplicamos un escalar por un arreglo, lo que nos devuelve un arreglo.
- 3 Cuando multiplicamos un arreglo por un arreglo, que nos devuelve un arreglo.

# Uso de la función `dot`

Entonces la manera en que debemos de usar la función `dot` es la siguiente:

```
>>> dot(L,U)
array([[ 8., -6.,  2.],
       [-4., 11., -7.],
       [ 4., -7.,  6.]])
```

Que corresponde a la matriz inicial  $A$ .



# Solución para $L$

Resolvemos primero  $Ly = b$  con sustitución hacia adelante:

$$y_1 = 28$$

$$-\frac{y_1}{2} + y_2 = -40 \quad y_2 = -40 + \frac{y_1}{2} = -26$$

$$\frac{y_1}{2} - \frac{y_2}{2} + y_3 = 33 \quad y_3 = 33 + \frac{y_2}{2} + \frac{y_1}{2} = 6$$

# Solución para $U$

La solución  $x$  ahora se obtiene de  $Ux = y$  por sustitución hacia atrás:

$$2x_3 = y_3 \quad x_3 = y_3/2 = 3$$

$$8x_2 - 6x_3 = y_2 \quad x_2 = \frac{y_2 + 6x_3}{8} = -1$$

$$8x_1 - 6x_2 + 2x_3 = y_1 \quad x_1 = \frac{y_1 + 6x_2 - 2x_3}{8} = 2$$

# Solución para $U$

La solución  $x$  ahora se obtiene de  $Ux = y$  por sustitución hacia atrás:

$$2x_3 = y_3 \quad x_3 = y_3/2 = 3$$

$$8x_2 - 6x_3 = y_2 \quad x_2 = \frac{y_2 + 6x_3}{8} = -1$$

$$8x_1 - 6x_2 + 2x_3 = y_1 \quad x_1 = \frac{y_1 + 6x_2 - 2x_3}{8} = 2$$

Por tanto la solución es  $x = [2, -1, 3]^T$

1. Introducción

2. Sistemas lineales

3. Sistemas algebraicos

4. Métodos de solución

5. Tres métodos directos

6. Método de eliminación de Gauss

6.1 Descripción del método

6.2 Fase de eliminación

# Método de eliminación de Gauss

La eliminación de Gauss es el método más conocido para la solución de ecuaciones simultáneas. Se compone de dos partes:

- 1 la fase de eliminación.

# Método de eliminación de Gauss

La eliminación de Gauss es el método más conocido para la solución de ecuaciones simultáneas. Se compone de dos partes:

- 1 la fase de eliminación.
- 2 la fase de solución.

# Fase de eliminación

La función de la fase de eliminación es transformar las ecuaciones en la forma  $Ux = c$ .

Las ecuaciones son resueltas por sustitución hacia atrás.

# Fase de eliminación

Con el fin de ilustrar el procedimiento, vamos a resolver las ecuaciones

$$4x_1 - 2x_2 + x_3 = 11 \quad (1)$$

$$-2x_1 + 4x_2 - 2x_3 = -16 \quad (2)$$

$$x_1 - 2x_2 + 4x_3 = 17 \quad (3)$$



# Fase de eliminación

La fase de eliminación utiliza solamente una de las operaciones elementales: multiplicar una ecuación (por ejemplo, ecuación  $j$ ) por una constante  $\lambda$  y restarla de otra ecuación (ecuación  $i$ ).

La representación simbólica de esta operación es

$$Eq.(i) \leftarrow Eq.(i) - \lambda \times Eq.(j)$$

La ecuación que se resta, a saber, la  $Eq.(j)$ , se llama ecuación pivote.

# Fase de eliminación

Comenzamos la eliminación eligiendo la Eq.(1) como ecuación pivote y eligiendo los multiplicadores  $\lambda$  a fin de eliminar  $x_1$  a partir de las Ecs. (2) y (3):

$$Eq.(2) \leftarrow Eq.(2) - (-0.5) \times Eq.(1)$$

$$Eq.(3) \leftarrow Eq.(3) - 0.25 \times Eq.(1)$$

# Fase de eliminación

Después de esta transformación, las ecuaciones quedan:

$$4x_1 - 2x_2 + x_3 = 11 \quad (4)$$

$$3x_2 - 1.5x_3 = -10.5 \quad (5)$$

$$-1.5x_2 + 3.75x_3 = 14.25 \quad (6)$$

Esto completa el primer paso.

Ahora elegimos  $Eq.(5)$  como ecuación pivote y eliminamos  $x_2$  de la  $Eq.(6)$ :

$$Eq.(6) \leftarrow Eq.(6) - (-0.5) \times Eq.(5)$$

que nos devuelve

$$4x_1 - 2x_2 + x_3 = 11 \quad (7)$$

$$3x_2 - 1.5x_3 = -10.5 \quad (8)$$

$$3x_3 = 9 \quad (9)$$

# Fase de eliminación

La fase de eliminación está completa. Las ecuaciones iniciales fueron re-emplazadas por un conjunto de ecuaciones equivalentes que pueden resolverse fácilmente por sustitución hacia atrás.

# Uso de la matriz aumentada

Como se ha señalado antes, la matriz de coeficientes aumentada es un instrumento más conveniente para realizar los cálculos.

Así, las ecuaciones originales se escribirían

$$\left[ \begin{array}{ccc|c} 4 & -2 & 1 & 11 \\ -2 & 4 & -2 & -16 \\ 1 & -2 & 4 & 17 \end{array} \right]$$

# Sistema obtenido

Las ecuaciones equivalentes producidas por el primer y segundo paso de la eliminación de Gauss, quedarían como:

$$\left[ \begin{array}{ccc|c} 4 & -2 & 1 & 11.00 \\ 0 & 3 & -1.5 & -10.50 \\ 0 & -1.5 & 3.75 & 14.25 \end{array} \right]$$

$$\left[ \begin{array}{ccc|c} 4 & -2 & 1 & 11.00 \\ 0 & 3 & -1.5 & -10.50 \\ 0 & 0 & 3 & 9.00 \end{array} \right]$$

# Fase de sustitución hacia atrás

Las incógnitas ahora puede ser calculadas por sustitución hacia atrás. Resolviendo las ecuaciones (9), (8) y (7) en ese orden, se obtiene

$$x_3 = 9/3 = 3$$

$$x_2 = (-10.5 + 1.5x - 3)/3 = -2$$

$$x_1 = (11 + 2x_2 - x_3)/4 = 1$$



# Algoritmo para la eliminación de Gauss

Para codificar el algoritmo de la fase de eliminación del método de eliminación.

Supongamos que las primeras  $k$  filas de  $A$  ya se han transformado a una forma triangular superior.

Por lo que la ecuación pivote actual es la  $k$ -ésima ecuación, y todas las ecuaciones de abajo están para ser transformadas.

# Matriz aumentada

$$\left[ \begin{array}{cccccccc|c} A_{11} & A_{12} & A_{13} & \dots & A_{1k} & \dots & A_{ij} & \dots & A_{1n} & b_1 \\ 0 & A_{22} & A_{23} & \dots & A_{2k} & \dots & A_{2j} & \dots & A_{2n} & b_2 \\ 0 & 0 & A_{33} & \dots & A_{3k} & \dots & A_{3j} & \dots & A_{3n} & b_3 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & A_{kk} & \dots & \dots & A_{kj} & \dots & A_{kn} & b_k \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & A_{ik} & \dots & \dots & A_{ij} & \dots & A_{in} & b_i \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & A_{nk} & \dots & \dots & A_{nj} & \dots & A_{nn} & b_n \end{array} \right]$$

# Operaciones elementales

Sea la  $i$ -ésima fila una fila por debajo de la ecuación pivote que será transformada, lo que significa que el elemento  $A_{ik}$  se va a eliminar.

Logramos esto mediante la multiplicación de la fila pivote por  $\lambda = A_{ik}/A_{kk}$  y restarlo de la fila  $i$ -ésima. Los correspondientes cambios en la fila  $i$  son

$$\begin{aligned}A_{ij} &\leftarrow A_{ij} - \lambda A_{kj}, & j = k, k+1, \dots, n \\ b_i &\leftarrow b_i - \lambda b_k\end{aligned}$$

# Operaciones elementales

Para transformar la matriz de coeficientes a una forma triangular superior, los índices  $k$  e  $i$  en las ecuaciones anteriores debe tener los intervalos  
 $k = 1, 2, \dots, n - 1$  (de la fila pivote),  
 $i = k + 1, k + 2, \dots, n$  (de la fila que se va a transformar)

# Algoritmo para la eliminación de Gauss

El algoritmo para la fase de eliminación ahora se escribe:

## Código 2: Eliminación de Gauss

```
1 for k in range(0, n - 1):  
2     for i in range(k + 1, n):  
3         if a[i, k] == 0.0:  
4             lam = a[i, k]/a[k, k]  
5             a[i, k+1:n] = a[i, k+1:n]  
6             ] - lam*a[k, k+1:n]  
             b[i] = b[i] - lam*b[k]
```

# Consideraciones

Con el fin de evitar operaciones innecesarias, el algoritmo anterior considera lo siguiente:

- 1 Si  $A_{ik}$  vale cero, la transformación de la fila  $i$  se omite.

# Consideraciones

Con el fin de evitar operaciones innecesarias, el algoritmo anterior considera lo siguiente:

- 1 Si  $A_{ik}$  vale cero, la transformación de la fila  $i$  se omite.
- 2 El índice  $j$  en la ecuación de transformación comienza con  $k + 1$  en lugar de  $k$ .

Por lo tanto,  $A_{ik}$  no se re-emplaza con cero, pero conserva su valor original.

Como la fase de solución nunca accede a la parte triangular inferior de la matriz de coeficientes, su contenido es irrelevante.

# Fase de sustitución

Luego de la eliminación de Gauss, la matriz de aumentada tiene la siguiente forma

$$\left[ A \mid b \right] = \left[ \begin{array}{ccccc|c} A_{11} & A_{12} & A_{13} & \dots & A_{1n} & b_1 \\ 0 & A_{22} & A_{23} & \dots & A_{2n} & b_2 \\ 0 & 0 & A_{33} & \dots & A_{3n} & b_3 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & A_{nn} & b_n \end{array} \right]$$



# Fase de sustitución

La última ecuación  $A_{nn}x_n = b_n$  se resuelve primero para obtener

$$x_n = \frac{b_n}{A_{nn}}$$

# Fase de sustitución

Consideremos ahora la etapa de sustitución hacia atrás donde  $x_n, x_{n-1}, \dots, x_{k+1}$  ya han sido calculados (en ese orden), para determinar  $x_k$  de la  $k$ -ésima ecuación:

$$A_{kk}x_k + A_{k,k+1}x_{k+1} + \dots + A_{kn}x_n = b_k$$

# Fase de sustitución

La solución es:

$$x_k = \left( b_k - \sum_{j=k+1}^n A_{kj} x_j \right) \frac{1}{A_{kk}}, \quad k = n-1, n-2, \dots, 1$$

Con Python tendremos que:

Código 3: Fase de sustitución

```
1 for k in range(n-1, -1, -1):  
2     b[k] = (b[k] - dot(a[k, k+1:n], b[  
    k+1:n])) / a[k, k]  
3 return b
```

# Ejercicio

Completa tu código para resolver  $Ax = b$ , donde

$$A = \begin{bmatrix} 2 & -3 & -1 \\ 3 & 2 & -5 \\ 2 & 4 & -1 \end{bmatrix} \quad b = \begin{bmatrix} 3 \\ -9 \\ -5 \end{bmatrix}$$

# Código para el ejercicio I

## Código 4: Función gaussElimin

```
1 def gaussElimin(a,b):
2     n = len(b)
3
4     for k in range(0, n - 1):
5         for i in range(k + 1,n):
6             if a[i, k] != 0.0:
7                 lam = a[i, k]/a[k, k]
8                 a[i, k+1:n] = a[i, k
+1:n] - lam*a[k, k+1:n]
```

## Código para el ejercicio II

```
9             b[i] = b[i] - lam*b[
10 k]
11     for k in range(n - 1, -1, -1):
12         b[k] = (b[k] - dot(a[k, k+1:
13 n], b[k+1:n]))/a[k,k]
14     return b
```

# Agregando los arreglos

## Código 5: Agregando los arreglos

```
1 a = array([[2., -3, -1], [3, 2, -5], [2,  
    4, -1]])  
2 b = array([3., -9, -5])  
3 x = gaussElimin(a,b)  
4  
5 print (x)
```

# Corroborando el resultado con `solve`

Podemos usar `linalg.solve` para verificar nuestro resultado, hacemos entonces:

```
1 print (linalg.solve(a,b))
```



# Corroborando el resultado con `solve`

Podemos usar `linalg.solve` para verificar nuestro resultado, hacemos entonces:

```
1 print (linalg.solve(a,b))
```

Pero nos damos cuenta de que el resultado es completamente diferente, ¿qué es lo que pasa?

# Respuesta

Lo que ocurre es que al llamar la función `gaussElimin(a, b)`, se modifican los valores de las entradas al llevarse a cabo las operaciones elementales, por lo que los arreglos que se usan para la función `linalg.solve(a, b)`, ya son otros arreglos.

Para corregir esta situación, debemos de garantizar que los arreglos a usar en `linalg.solve`, son idénticos a los arreglos iniciales, para ello, hacemos una copia:

### Código 6: Ajuste a la solución

```
1 a = array([[2., -3, -1], [3, 2, -5], [2,  
    4, -1]])  
2 b = array([3., -9, -5])  
3  
4 c = a.copy()  
5 d = b.copy()  
6  
7 print (gaussElimin(a,b))  
8 print (linalg.solve(c,d))
```

Y vemos que ambos resultados son los mismos.

# Conjuntos de ecuaciones múltiples

Como hemos mencionado, a menudo es necesario para resolver las ecuaciones  $A\mathbf{x} = \mathbf{b}$  para varios vectores constantes.

Sea  $m$  dichos vectores constantes, denotados por  $\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_m$  y dejar que los correspondientes vectores solución sean  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m$ .

# Conjuntos de ecuaciones múltiples

Escribimos el conjunto de ecuaciones múltiples  $\mathbf{AX} = \mathbf{B}$ , donde

$$\mathbf{X} = [\mathbf{x}_1 \ \mathbf{x}_2 \ \dots \ \mathbf{x}_m]$$

$$\mathbf{B} = [\mathbf{b}_1 \ \mathbf{b}_2 \ \dots \ \mathbf{b}_m]$$

son matrices de  $n \times m$  cuyas columnas consisten en vectores solución y vectores constantes, respectivamente.

# Conjuntos de ecuaciones múltiples

Una manera económica de manejar ecuaciones de este tipo durante la fase de eliminación es incluir todos los  $m$  vectores constantes en la matriz de coeficientes aumentada, de modo que se transforman simultáneamente con la matriz de coeficientes.

# Conjuntos de ecuaciones múltiples

Las soluciones son luego obtenidas por sustitución hacia atrás de la manera habitual, un vector a la vez.

Tendremos que ajustar el método de Gauss para realizar esta tarea.



# Ejercicio

Resolver el conjunto de ecuaciones  $AX = B$ ,  
donde:

$$A = \begin{bmatrix} 2 & 0 & -1 & 0 \\ 0 & 1 & 2 & 0 \\ -1 & 2 & 0 & -2 \\ 0 & 0 & 1 & -2 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$$