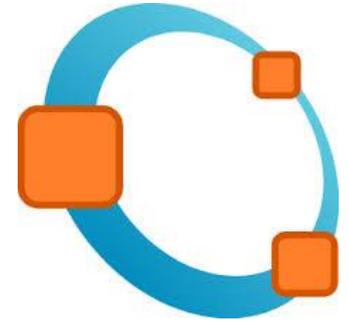


Introducción a Octave

Unidad 3



Daniel Millán
Nora Moyano & Iván Ferrari

San Rafael, Argentina 2018



Departamento de
Ingeniería Mecánica



UNCUYO
UNIVERSIDAD
NACIONAL DE CUYO



FACULTAD DE
**CIENCIAS APLICADAS
A LA INDUSTRIA**



Funciones de biblioteca

- Octave tiene un gran número de funciones incorporadas, cuyos aspectos más relevantes describiremos en este doc.
- Ciertas funciones vienen incorporadas en el propio código ejecutable y son particularmente rápidas y eficientes.
- Existen además funciones definidas en ficheros ***.m** y ***.oct** que vienen con el propio programa o que han sido aportadas por usuarios del mismo.
- Los archivos **oct** son piezas de código C++ que se han compilado con la API **Octave** (Application Programming Interface).
- Octave proporciona el comando **mkoctfile** para construir archivos **oct** a partir de código fuente en C, C++ o Fortran.
- Para que Octave encuentre una determinada función de usuario dicho archivo-M debe estar en el directorio actual de trabajo o en el **search path**.



Funciones de biblioteca

- Octave incluye una interfaz para permitir el uso de archivos ***.mex** y para compartir código compilado entre Octave y MATLAB.
- Dado que los ***.mex** emplean subrutinas, funciones y procedimientos de la estructura interna de MATLAB (diferentes de Octave), un archivo ***.mex** nunca poseerá el mismo rendimiento en Octave que el archivo ***.oct** equivalente.
- Por ej. cuando se invoca una función **mex-file**, para pasar las variables a las funciones **mex** se emplean un número significativo de copias adicionales de bloques de memoria.
- Se recomienda que cualquier nuevo código en C, C++ o Fortran se escriba con la interfaz **oct-file** para generar el archivo **oct** mediante la orden **mkoctfile**.



Funciones de biblioteca

1. Características generales de las funciones de Octave.
2. Funciones matemáticas elementales que operan de modo escalar y que actúan sobre vectores/matrices.
3. Funciones matriciales elementales y especiales.
4. Factorización y/o descomposición matricial.
5. Más operadores relacionales vectores/matrices.
6. Otras funciones que actúan sobre vectores/matrices.
7. Funciones para cálculos con polinomios.



1. Características generales de las funciones de Octave.

- Una función tiene **nombre**, **valor de retorno** y **argumentos**.
- Una función se “llama” utilizando su nombre en una expresión o utilizándolo como un comando más.
- Las funciones se pueden definir en ficheros de texto ***.m**

Ejemplo: Llamadas a funciones

```
>> [maximo, posmax] = max(x);  
>> r = sqrt(x2+y2) + eps;  
>> a = cos(alfa) - sin(alfa);
```

- ✓ Los **nombres** de las funciones se han puesto en negrita.
- ✓ Los **argumentos** de cada función van a continuación del nombre entre paréntesis (y separados por comas si hay más de uno).
- ✓ Los **valores de retorno** son el resultado de la función y sustituyen a ésta en la expresión donde la función aparece.



1. Características generales de las funciones de Octave.

- En Octave las funciones pueden tener ***valores de retorno matriciales múltiples***.
 - Los valores de retorno ***se recogen entre corchetes***, separados por comas.
 - Las funciones que no tienen argumentos no llevan paréntesis, por lo que a simple vista no siempre son fáciles de distinguir de las variables.
- Ejemplo:** *eps* es una función sin argumentos ¿otras?
- Los nombres de las funciones ***NO son palabras reservadas***.
 - Por ej. es posible crear una variable llamada ***sin*** o ***cos***, que oculte las funciones correspondientes.
 - Para poder acceder a las funciones originales hay que eliminar (***clear***) las variables del mismo nombre que las ocultan (visibles en el Espacio de Trabajo).



1. Características generales de las funciones de Octave.

- En Octave hay diversos tipos de funciones. Los tipos de funciones más importantes, clasificadas según su finalidad:
 - Funciones matemáticas elementales.
 - Funciones especiales.
 - Funciones matriciales elementales.
 - Funciones matriciales específicas.
 - Funciones para descomposición y/o factorización de matrices.
 - Funciones para análisis estadístico de datos.
 - Funciones para manejo de conjunto de datos.
 - Funciones para análisis de polinomios e interpolación.
 - Funciones para resolución de ecs. diferenciales ordinarias.
 - Resolución de ecuaciones no-lineales y optimización.
 - Integración numérica.
 - Funciones para procesamiento de señales, sonido e imagen.
 - Geometría (qhull library).



1. Características generales de las funciones de Octave.

- Características generales de las funciones de Octave:
 - Los **argumentos** de las funciones pueden ser expresiones y llamadas a otra función.
 - Las funciones no modifican las variables que se pasan como argumento, a no ser que se incluyan también como valores de retorno. Se pasan **por valor, no por referencia**.
 - Octave admite valores de retorno matriciales múltiples.
 - >> `[V, D] = eig(A)` %vectores y valores propios de A.
 - >> `[xmax, imax] = max(x)`
 - >> `xmax = max(x)`
 - Suma/resta de una matriz con un escalar consisten en sumar/restar el escalar a todos los elementos de la matriz.
 - Tecleando **help nombre_funcion** se obtiene información sobre la función de ese nombre.



2. Funciones matemáticas elementales que operan de modo **escalar**

- Estas comprenden las funciones matemáticas trascendentales y otras funciones básicas.
- Cuando se aplican a una matriz actúan sobre cada elemento de la matriz como si se tratase de un escalar.
- Por tanto, se aplican de la misma forma a escalares, vectores y matrices.
- Algunas de las funciones de este grupo son las siguientes:
 - **sin**(x) seno, **cos**(x) coseno, **tan**(x) tangente
 - **asin**(x) arco seno, **acos**(x) arco coseno
 - **atan**(x) arco tangente (ángulo entre $-\pi/2$ y $+\pi/2$)
 - **atan2**(y,x) arco tangente (ángulo entre $-\pi$ y $+\pi$); se le pasan 2 argumentos, proporcionales al seno y coseno
 - **sinh**(x), **cosh**(x), **tanh**(x) seno, coseno y tan hiperbólico
 - **log**(x) logaritmo natural, **exp**(x) función exponencial



2. Funciones que operan sobre vectores.

- Las siguientes funciones **sólo actúan sobre vectores**:
 - **min**(x)/**max**(x) mínimo/máximo elemento de un vector
 - **sum**(x) suma de los elementos de un vector
 - **cumsum**(x) devuelve el vector suma acumulativa de los elementos de un vector
 - **mean**(x) valor medio de los elementos de un vector
 - **std**(x) desviación típica
 - **prod**(x) producto de los elementos de un vector
 - **cumprod**(x) devuelve el vector producto acumulativo de los elementos de un vector
 - $[y,i]=$ **sort**(x) ordenación ascendente de los elementos de **x**
- En realidad estas funciones **se pueden aplicar también a matrices**, pero en ese caso **se aplican por separado a cada columna de la matriz**, dando como valor de retorno un vector resultado.

3. Funciones que operan sobre matrices.

- **Funciones matriciales elementales:**
 - $B = A'$ calcula la traspuesta (conjugada) de la matriz **A**
 - $B = A.'$ calcula la traspuesta (sin conjugar) de la matriz **A**
 - $v = \text{poly}(A)$ devuelve un vector **v** con los coeficientes del polinomio característico de la matriz cuadrada **A**
 - $t = \text{trace}(A)$ devuelve la traza **t** (suma de los elementos de la diagonal) de una matriz cuadrada **A**
 - $[m,n] = \text{size}(A)$ devuelve el número de filas **m** y de columnas **n** de una matriz rectangular **A**
 - $n = \text{size}(A)$ devuelve el tamaño de una matriz cuadrada **A**
 - $nf = \text{size}(A,1)$ devuelve el número de filas de **A**
 - $nc = \text{size}(A,2)$ devuelve el número de columnas de **A**

3. Funciones que operan sobre matrices.

- **Funciones matriciales especiales:**
- Las funciones ***exp()***, ***sqrt()*** y ***log()*** se aplican ***elemento a elemento*** a las matrices y/o vectores.
- Existen otras funciones similares que se aplican a una matriz como una única entidad:
 - **expm(A)** si $A=XD X'$ $\Leftrightarrow \text{expm}(A) = X * \text{diag}(\text{exp}(\text{diag}(D))) * X'$
 - **sqrtn(A)** devuelve una matriz que multiplicada por sí misma da la matriz **A**
 - **logm()** es la función recíproca de expm(A)
- Aunque no pertenece a esta familia de funciones, el ***operador potencia*** (**^**) está emparentado con ellas:
 - A^n está definida si **A** es cuadrada y **n** un número real.
 - Si **n** es entero, el resultado se calcula por multiplicaciones sucesivas.
 - Si **n** es real, el resultado se calcula como: $A^n = X * D.^n * \text{inv}(X)$ siendo $[X,D]=\text{eig}(A)$. **Ejemplo:** Verifique $2^A = X * 2^D / X$.



4. Funciones de factorización y/o descomposición matricial.

1) Basadas en factorización triangular (eliminación de Gauss)

- $U = \text{chol}(A)$ descomposición de Cholesky de matriz simétrica y definida positiva. El resultado es una matriz U triangular superior tal que $A = U' * U$
- $[L,U] = \text{lu}(A)$ descomposición de Crout ($A = LU$). La matriz L es una permutación de una matriz triangular inferior
- $B = \text{inv}(A)$ calcula la inversa de A . Equivale a $B = \text{inv}(U) * \text{inv}(L)$
- $d = \text{det}(A)$ devuelve el determinante d de la matriz cuadrada A . Equivale a $d = \text{det}(L) * \text{det}(U)$
- $[E,xc] = \text{rref}(A)$ reducción a forma de escalón con un vector xc que da una posible base del espacio de columnas de A
- $c = \text{rcond}(A)$ devuelve una estimación del recíproco de la condición numérica de la matriz A basada en la norma-1. Si el resultado es próximo a 1 la matriz A está bien condicionada; si es próximo a 0 no lo está.



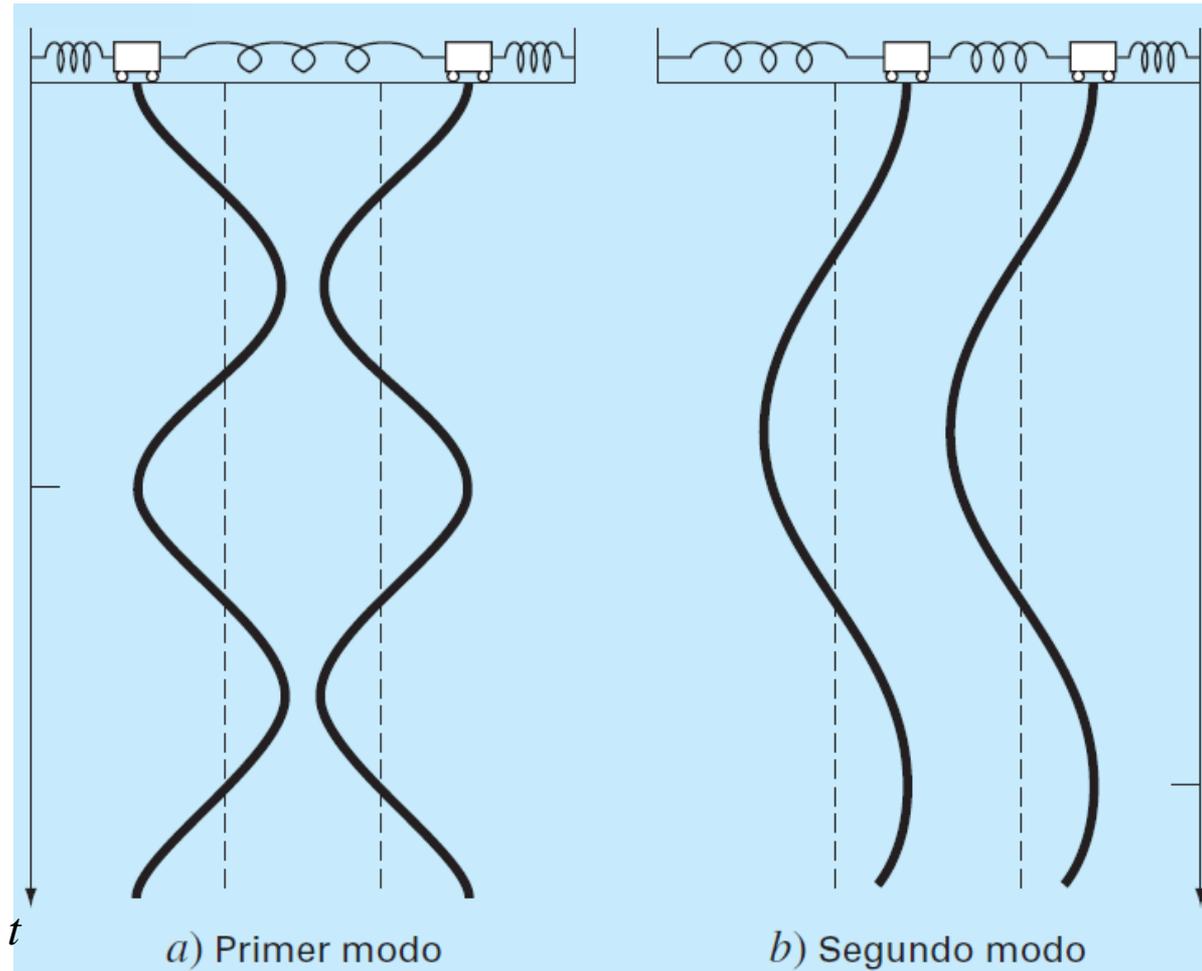
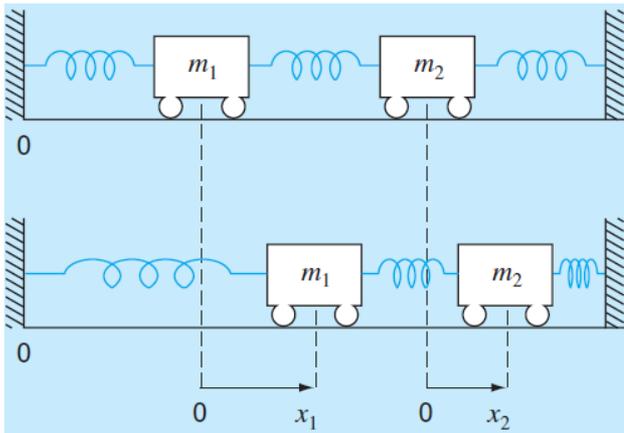
4. Funciones de factorización y/o descomposición matricial.

2) Basadas en el cálculo de valores y vectores propios

- El cálculo de los valores propios y de los vectores propios de una matriz simétrica tiene gran importancia en las matemáticas y en la ingeniería.
- Los problemas de *valores propios*, o característicos o eigenvalores, constituyen una clase especial de problemas con valores en la frontera, que son comunes en el contexto de problemas de ingeniería que implican vibraciones, elasticidad y otros sistemas oscilantes.
- Cabe destacar, el problema de la diagonalización de una matriz, el cálculo de los momentos de inercia y de los ejes principales de inercia de un sólido rígido, o de las frecuencias propias de oscilación de un sistema oscilante.

4. Funciones de factorización y/o descomposición matricial.

2) Basadas en el cálculo de valores y vectores propios



Chapra y Canale, Métodos Numéricos Para Ingeniería, Cap.27, 5ta Ed, 2007.



4. Funciones de factorización y/o descomposición matricial.

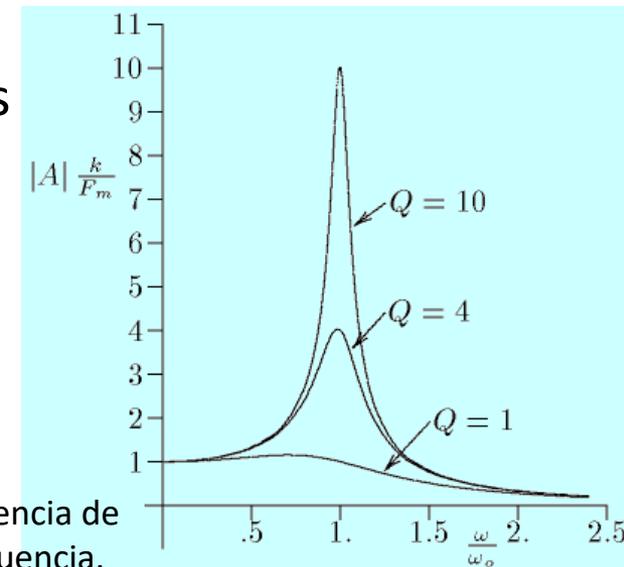
2) Basadas en el cálculo de valores y vectores propios

- $[X, D] = \text{eig}(A)$ vectores propios (columnas de X) y valores propios (diagonal de D) de una matriz cuadrada A . Con frecuencia el resultado posee números complejos si A no es simétrica.
- $[X, D] = \text{eig}(A, B)$ vectores propios (columnas de X) y valores propios (diagonal de D) de dos matrices cuadradas A y B ($Ax = \lambda Bx$).

Los vectores propios están normalizados de modo que $X' * B * X = I$.

Cuando A es simétrica y B es simétrica y definida positiva se puede utilizar $[X, D] = \text{eig}(A, B, 'chol')$.

Wiki. Respuesta en frecuencia de un oscilador armónico. A la frecuencia de resonancia, la amplitud es Q veces más grande que a muy baja frecuencia.

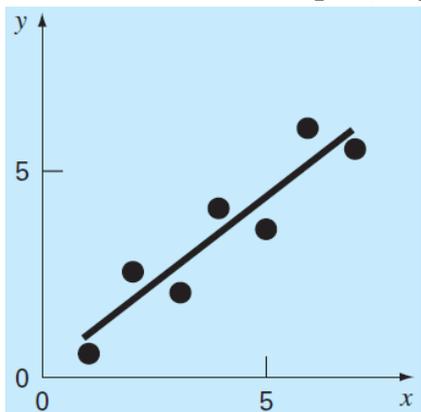




4. Funciones de factorización y/o descomposición matricial.

3) Basadas en la descomposición QR

- $[Q,R] = \text{qr}(A)$ descomposición $A=QR$ de una matriz rectangular.



Se utiliza para sistemas con más ecuaciones que incógnitas ($m>n$). Q es una matriz ortogonal ($Q^*Q=I$) aunque A no lo sea. R es una matriz triangular superior, elementos diagonales de R pueden no ser positivos.

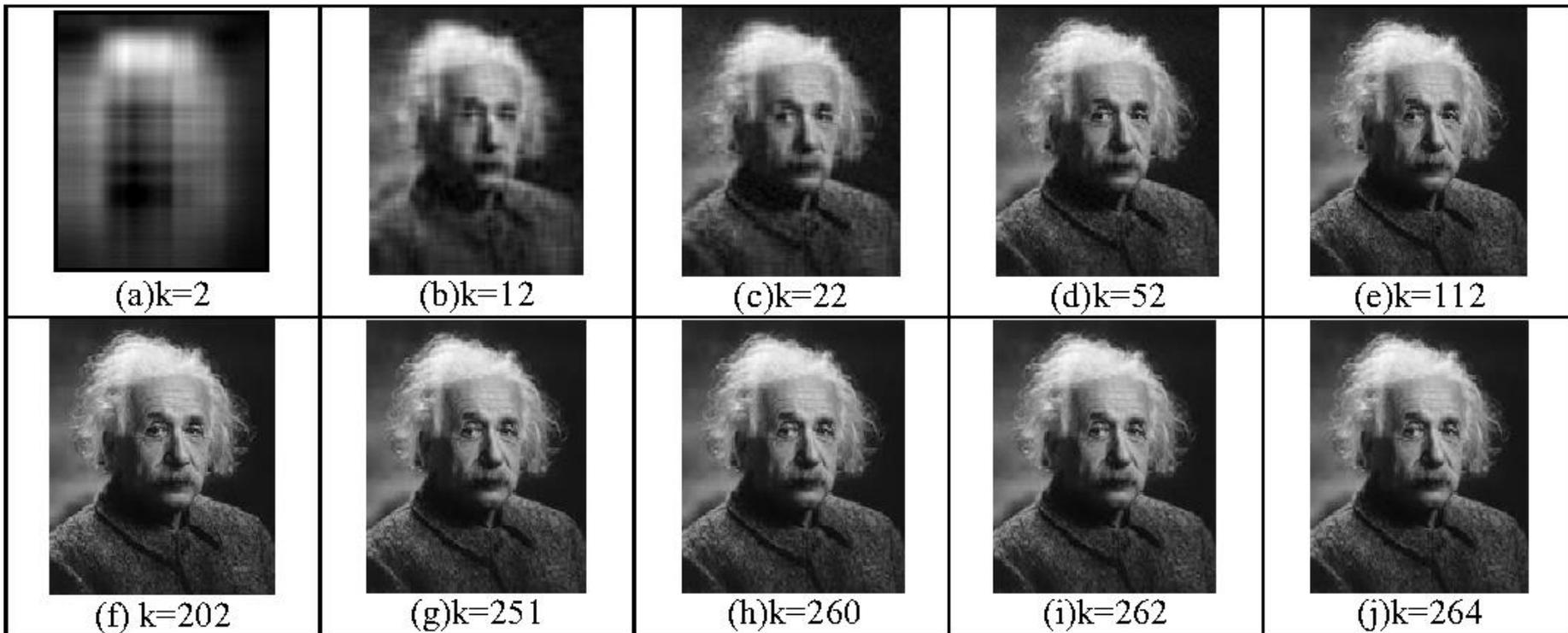
Se emplea para resolver **problemas de mínimos cuadrados**.

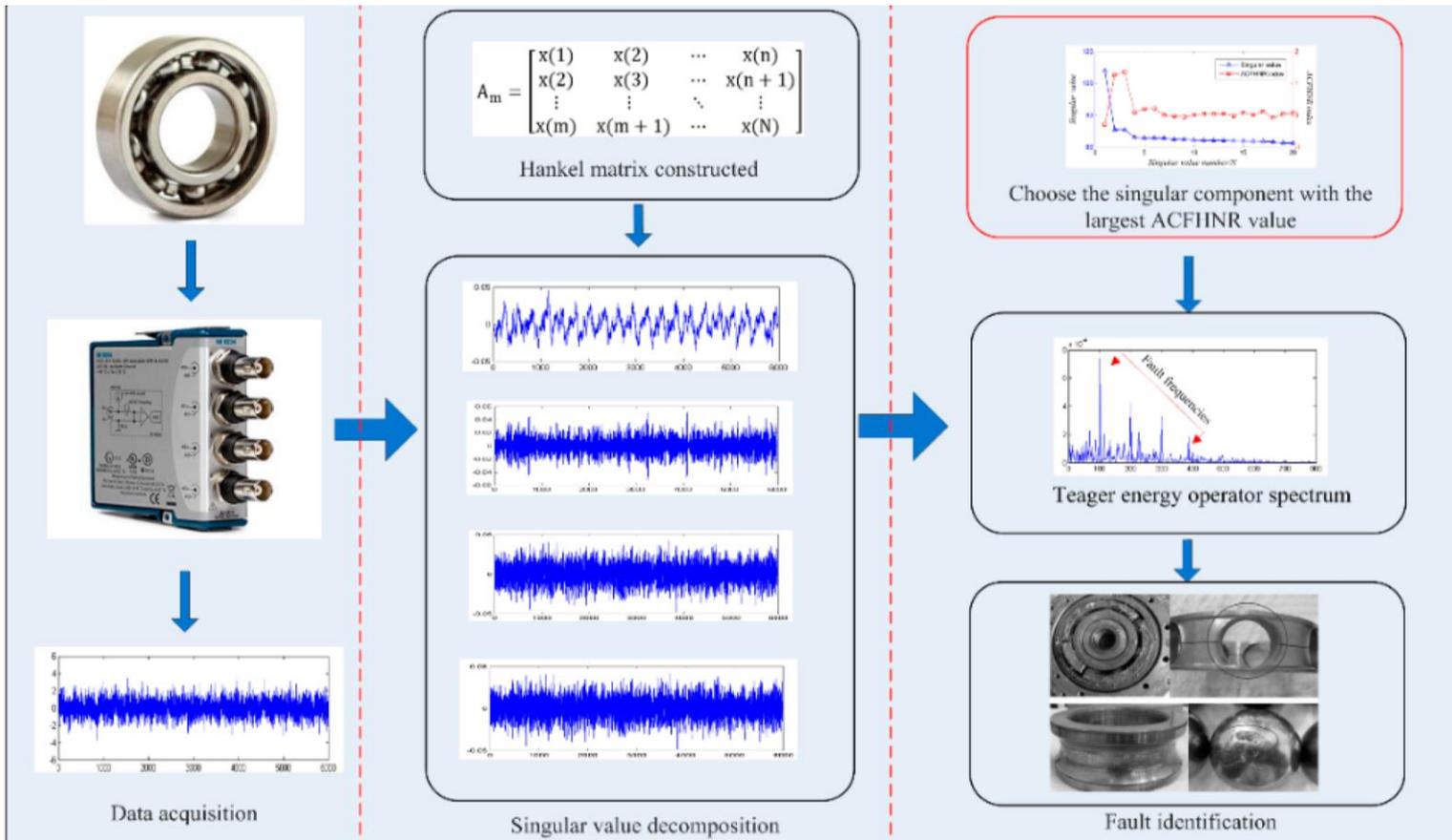
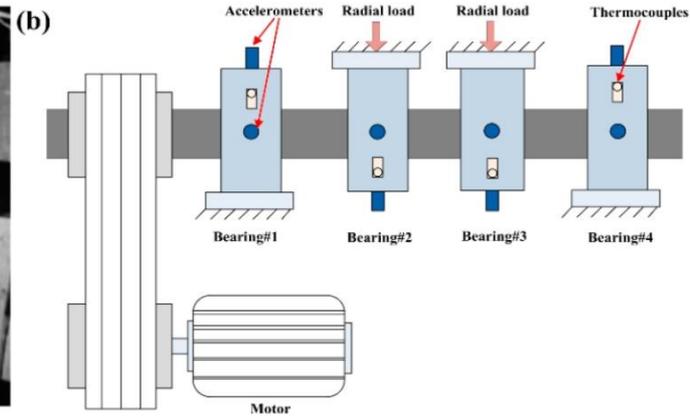
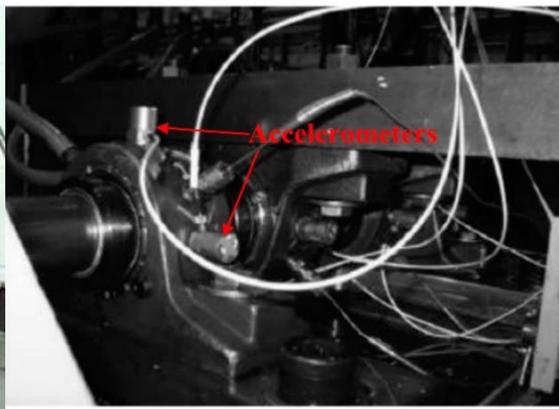
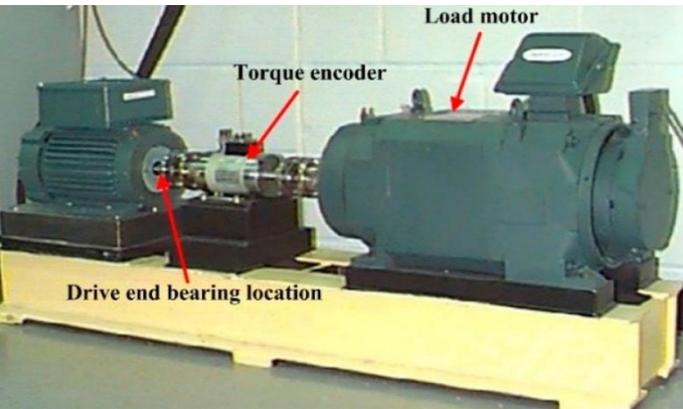
- $[Q,R,P]=\text{qr}(A)$ factorización QR con pivotamiento por columnas. La matriz P es una matriz de permutación tal que $A^*P=Q^*R$.
- $B = \text{null}(A)$ devuelve una base ortonormal del subespacio nulo ($Ax = 0$) de la matriz rectangular A .
- $Q = \text{orth}(A)$ devuelve una base ortonormal del espacio de columnas de A . El número de columnas de Q es el rango de A .

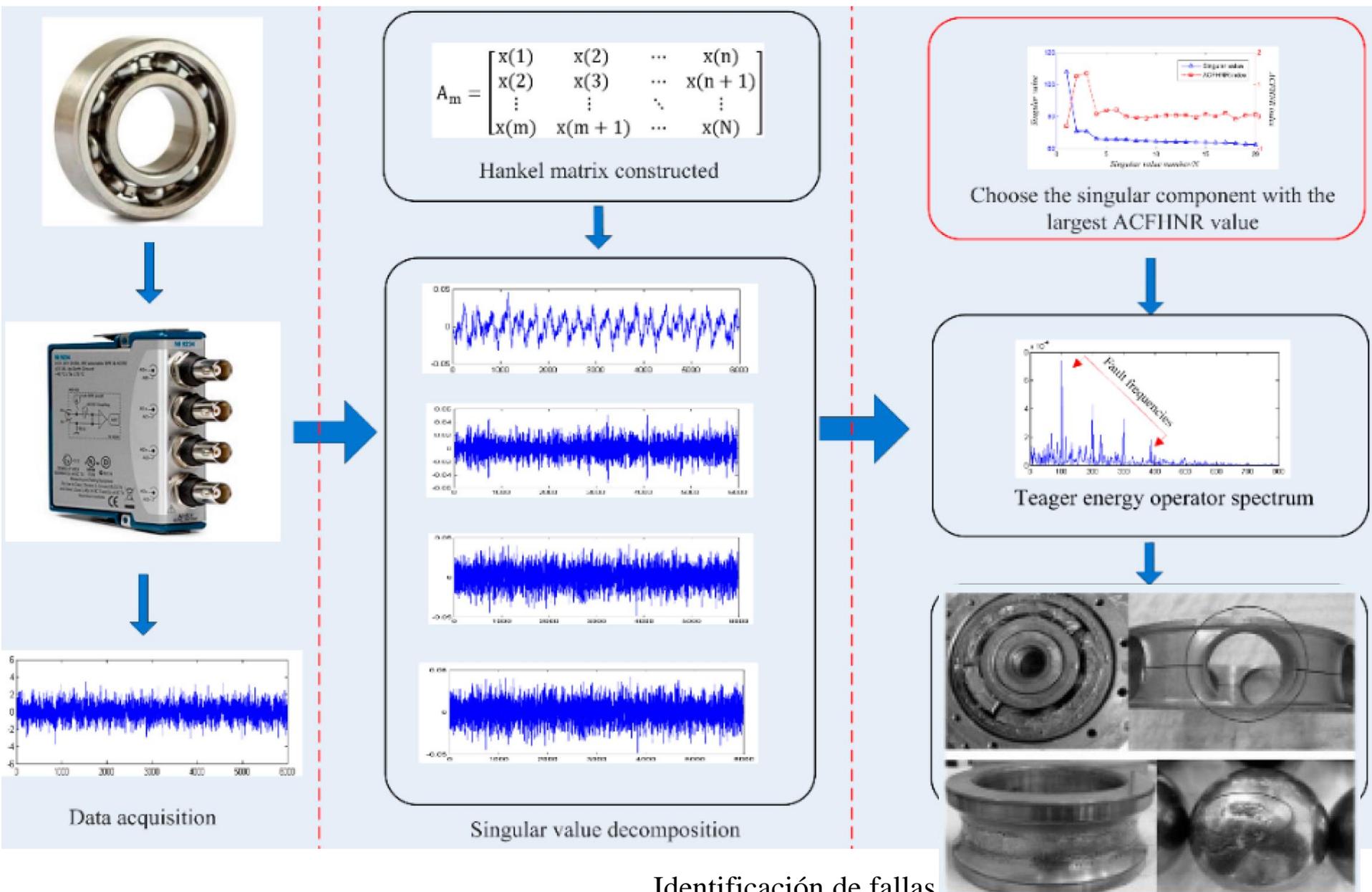
4. Funciones de factorización y/o descomposición matricial.

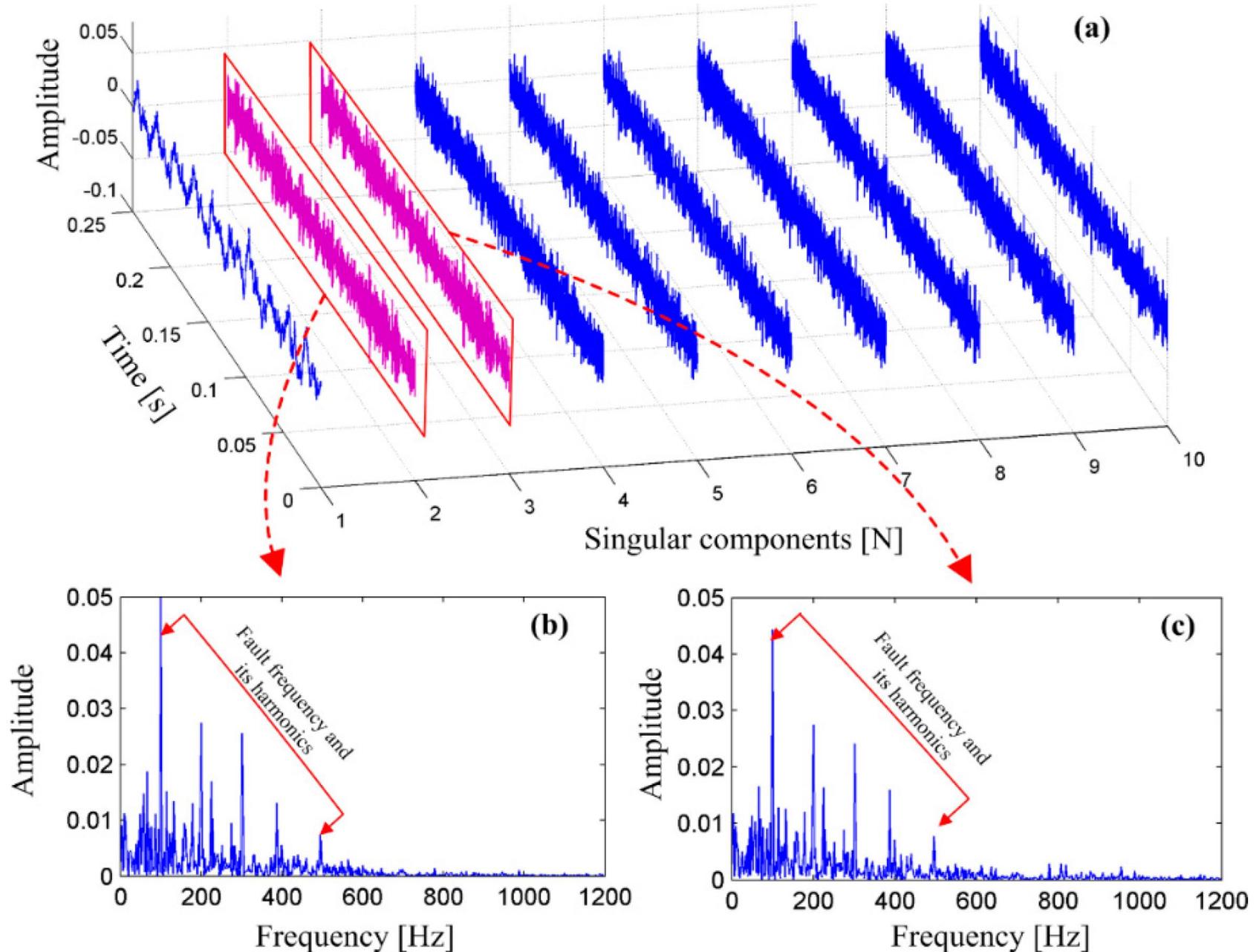
4) Basadas en la descomposición de valores singulares

- $[U, D, V] = \text{svd}(A)$ descomposición de valor singular de una matriz rectangular ($A=U \cdot D \cdot V'$). U y V son matrices ortonormales. D es diagonal y contiene los valores singulares.











4. Funciones de factorización y/o descomposición matricial.

4) Basadas en la descomposición de valores singulares

- $\text{nor} = \text{norm}(A)$ calcula la norma-2 de A (mayor valor singular).
- menos operaciones aritméticas que la función *norm*.
- $c = \text{cond}(A)$ condición numérica de la matriz A . Cociente entre el máx y el mín valor singular. $\text{cond}(A)$ da una idea del error que se obtiene al resolver un sistema de ecs lineales: $\log(\text{cond})$ indica el número de cifras significativas que se pierden.
- $r = \text{rank}(A)$ calcula el rango r de una matriz rectangular A .
- $B = \text{pinv}(A)$ calcula la pseudo-inversa de una matriz rectangular A .
- $\text{nor} = \text{normest}(A)$ calcula de forma aproximada la norma-2 con
- $c = \text{condest}(A)$ estimación por defecto de la condición numérica de A con la norma-1. Función mucho más económica que *cond*.



4. Funciones de factorización y/o descomposición matricial.

5) Cálculo del rango y normas

- El rango se calcula implícitamente (sin que el usuario lo pida) al ejecutar las funciones *rref(A)*, *orth(A)*, *null(A)* y *pinv(A)*. La función *rank(A)* está basada en *pinv(A)*. Con *pinv(A)* se utiliza la descomposición SVD, que es el método más fiable y más caro.
- Normas de vectores:
 - ✓ **norm**(x,p) norma-p, es decir $sum(abs(x)^p)^{1/p}$.
 - ✓ **norm**(x) norma-2 ó euclídea; equivale al módulo o *norm(x,2)*.
 - ✓ **norm**(x,inf) norma-∞, es decir $max(abs(x))$.
 - ✓ **norm**(x,1) norma-1, es decir $sum(abs(x))$.
- Normas de matrices:
 - **norm**(A) norma-2, máximo valor singular de **A**, $max(svd(A))$.
 - **normes**t(A) estimación de la norma-2. Útil para matrices grandes.
 - **norm**(A,1) norma-1 de **A**, máxima suma de valores absolutos por columnas, es decir: $max(sum(abs(A)))$.
 - **norm**(A,inf) norma-∞ de **A**, máxima suma de valores absolutos por filas, es decir: $max(sum(abs(A')))$.



5. Más sobre operadores relacionales: vectores/matrices

- Los operadores relacionales vistos previamente ($<$, $>$, $<=$, $>=$, $==$ y $\sim=$) actúa entre dos matrices/vectores del mismo tamaño, el resultado es otra matriz/vector de ese mismo tamaño conteniendo 1 y 0, (***true*** o ***false***).
- Las matrices "binarias" no se almacenan en memoria ni se asignan a variables, se procesan sobre la marcha. Octave dispone de varias funciones para ello. Cualquier valor $\neq 0$ equivale a ***true***, mientras que 0 equivale a ***false***.



5. Más sobre operadores relacionales: vectores/matrices

- Algunas de estas funciones son:
 - **any**(x) función vectorial; chequea si *alguno* de los elementos del vector **x** cumple una dada condición ($\neq 0$). Devuelve 1 ó 0.
 - **any**(A) se aplica por separado a cada columna de la matriz **A**. El resultado es un vector de unos y ceros.
 - **all**(x) función vectorial; chequea si *todos* los elementos del vector **x** cumplen una condición. Devuelve un 1 ó 0.
 - **all**(A) se aplica por separado a cada columna de la matriz **A**. El resultado es un vector de unos y ceros.
 - **find**(x) busca índices correspondientes a elementos de vectores que cumplen una determinada condición. El resultado es un vector con los índices que cumplen la condición.
 - **find**(A) cuando esta función se aplica a una matriz la considera como un vector con una columna detrás de otra, de la 1ª a la última, **A(:)**.



5. Más sobre operadores relacionales: vectores/matrices

- Veamos como funcionan algunas de ellos:

```
>> A=magic(3)
```

```
>> m=find(A>4)
```

```
>> A(m)=10*ones(size(m))
```

```
>> any(A==3)
```

```
>> any(ans)
```

```
>> all(all(A))
```

- La función *isequal(A, B)* devuelve *uno* si las matrices son idénticas y *cero* si no lo son.



6. Otras funciones que actúan sobre vectores/matrices

- Las siguientes funciones pueden actuar sobre vectores y matrices, y sirven para chequear ciertas condiciones:
 - **exist**('var') comprueba si el nombre **var** existe como variable, función, directorio, fichero, etc.
 - **isnan**(A) chequea si hay valores **NaN** en **A**, devolviendo una matriz de unos y ceros del mismo tamaño que **A**.
 - **isinf**(A) chequea si hay valores **Inf** en **A**, devolviendo una matriz de unos y ceros del mismo tamaño que **A**.
 - **isfinite**(A) chequea si los valores de **A** son finitos.
 - **isempty**(A) chequea si un vector o matriz está vacío o tiene tamaño nulo.
 - **ischar**() chequea si una variable es una cadena de caracteres.
 - **isglobal**() chequea si una variable es global.
 - **issparse**() chequea si una matriz es dispersa (*sparse*, es decir, con un gran número de elementos cero).



6. Otras funciones que actúan sobre vectores/matrices

- A continuación se presentan algunos ejemplos de uso de estas funciones en combinación con otras vistas previamente.
- Se define un vector x con un **NaN**, que se elimina en la forma:
 - >> $x=[1 \ 2 \ 3 \ 4 \ 0/0 \ 6]$
 - >> $i=find(isnan(x))$
 - >> $x=x(find(\sim isnan(x)))$
- Otras posibles formas de eliminarlo serían las siguientes:
 - >> $x=x(\sim isnan(x))$
 - >> $x(isnan(x))=[]$
- La siguiente sentencia elimina las filas de una matriz que contienen algún **NaN**:
 - >> $A(any(isnan(A)') , :)=[]$



7. Funciones para cálculos con polinomios

- Para Octave un polinomio se puede definir mediante un vector de coeficientes. Por ejemplo, el polinomio:

$$x^4 - 8x^2 + 6x - 10 = 0$$

se puede representar mediante el vector [1, 0, -8, 6, -10].

- MATLAB puede realizar diversas operaciones sobre él, como por ejemplo evaluarlo para un determinado valor de x (función ***polyval()***) y calcular las raíces (función ***roots()***):

```
>> pol=[1 0 -8 6 -10]
```

```
>> roots(pol)
```

```
>> polyval(pol,1)
```



7. Funciones para cálculos con polinomios

- Algunas funciones orientadas al cálculo con polinomios:
 - **poly**(A) polinomio característico de la matriz **A**
 - **roots**(pol) raíces del polinomio **pol**
 - **polyval**(pol,x) evaluación del polinomio **pol** para el valor de **x**.
Si **x** es un vector, **pol** se evalúa para cada elemento de **x**
 - **conv**(p1,p2) producto de convolución de dos polinomios **p1** y **p2**
 - [c,r]=**deconv**(p,q) división del polinomio **p** por el polinomio **q**.
En **c** se devuelve el cociente y en **r** el resto de la división
 - **residue**(p1,p2) descompone el cociente entre **p1** y **p2** en suma de fracciones simples (ver **>>help residue**)
 - **polyder**(pol) calcula la derivada de un polinomio
 - **polyder**(p1,p2) calcula la derivada de producto de polinomios
 - **polyfit**(x,y,n) calcula los coeficientes de un polinomio **p(x)** de grado **n** que se ajusta a los datos **p(x(i)) ≈ y(i)**, mínimo error cuadrático medio.
 - **interp1**(xp,yp,x) calcula el valor interpolado para la abscisa **x** a partir de un conjunto de puntos dado por los vectores **xp** e **yp**.