

 UNIVERSIDAD DE OVIEDO DEPARTAMENTO DE MATEMÁTICAS	Asignatura	Cálculo Numérico	Página 1 de 4
	Tema	Sistemas Lineales: Métodos directos	
	Práctica	5	
	Autor	César Menéndez Fernández	

1 Métodos directos

Comenzaremos viendo la coherencia de operaciones con matrices, y para ello realizaremos funciones simples que sumen o multipliquen matrices. Además utilizaremos esto como repaso de los temas anteriores (condicionales y funciones). También se empleará información sobre el número de argumentos de entrada, a la hora de verificar la coherencia de la operación. Para simplificar las operaciones, se introduce una nueva instrucción, **switch**, que ejecuta un conjunto de instrucciones seleccionadas a partir de los valores de una expresión. Cada alternativa se denomina “caso” y consta de la instrucción “case”, una o más expresiones y una o varias instrucciones. Su funcionamiento es similar al condicional, excepto que solo evalúa valores. Sintaxis:

```
switch expresión
case valor1_de_la_expresión
    instrucciones
case { valor2_de_la_expresión, valor3_de_la_expresión ,...}
    instrucciones
...
otherwise
    instrucciones
end
```

Ejemplo

```
method = 'Gauss';

switch lower(method)
case {'gauss','gauss-jordan'}
    disp('Método directo por sustitución')
case {'doolittle','crout','cholesky'}
    disp(' Método directo por factorización')
case {'jacobi','gauss-seidel','relajacion'}
    disp(' Método iterativo de primer orden')
otherwise
    disp(' Método desconocido')
end
```

Ejercicios recomendados:

1. Crear el programa opera.m que opera con matrices:

Entrada: variable, una o dos matrices dependiendo de la operación y el código de la operación. Se debe utilizar la sintaxis (matriz1, código_operación[,matriz2]). Los códigos de operación pueden ser: "+"(suma), "-"(resta), "*" (multiplicación) o "#"(inversa)

Salida: R, resultado de la operación

- Paso 1: Si hay dos variables (argumentos) de entrada
- Paso 2: Si la operación no es inversión, enviar mensaje de error y finalizar
- Paso 3: Obtener el numero de filas y columnas de la matriz1
- Paso 4: Si no es cuadrada, enviar mensaje de error y finalizar
- Paso 5: Si el rango no coincide con el orden, enviar mensaje de error y finalizar
- Paso 6: Calcular la inversa de la matriz1 y almacenarla en R
- Paso 7: Sino si hay 3 variables
- Paso 8: Obtener el numero de filas y columnas de la matriz1 y de la matriz2
- Paso 9: Caso de operación suma
- Paso 10: Si las filas o las columnas no coinciden, mensaje de error y finalizar
- Paso 11: Calcular la suma de la matriz1 con la matriz2 y almacenarla en R
- Paso 12: Caso de operación resta
- Paso 13: Si las filas o las columnas no coinciden, mensaje de error y finalizar
- Paso 14: Calcular la resta de la matriz1 con la matriz2 y almacenarla en R
- Paso 15: Caso de operación multiplicación
- Paso 16: Si las columnas de la matriz1 no coinciden con las filas de la matriz2,
- Paso 17: enviarmensaje de error y finalizar
- Paso 18: Calcular el producto de la matriz1 por la matriz2 y almacenarla en R
- Se debe comprobar el correcto funcionamiento probando con todos los casos posibles.

1.1 Resolución directa de sistemas lineales

La resolución de sistemas lineales es inmediata con MATLAB mediante la instrucción “\”. Así, para resolver el sistema $Ax=b$ es suficiente poner

» $x=A \backslash b$

Si bien también se habría podido obtener el resultado mediante

» $x=\text{inv}(A)*b$

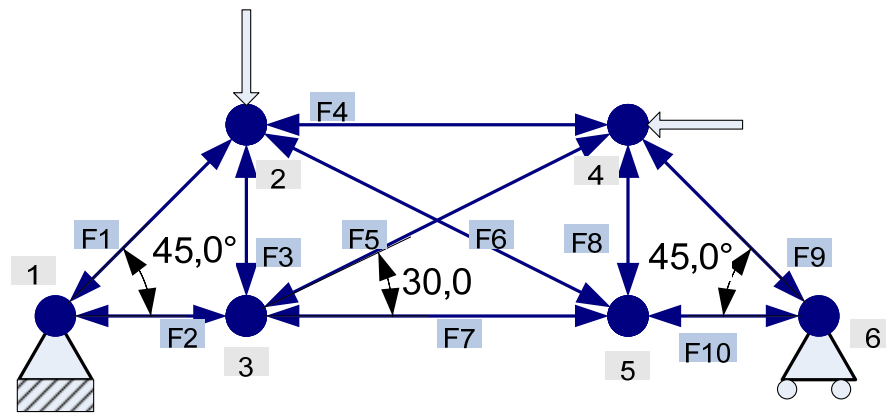
este modo se debe evitar al realizar el cálculo de una inversa, que es un problema más complejo.

Se pide:

2. Utilizar la instrucción \ para resolver el siguiente sistema

$$\begin{pmatrix} 112 & 84 & 48 & 28 \\ 84 & 81 & 77 & -7 \\ 48 & 77 & 146 & -83 \\ 28 & -7 & -83 & 85 \end{pmatrix} X = \begin{pmatrix} -204 \\ -186 \\ -193 \\ 38 \end{pmatrix}$$

3. Obtener los desplazamientos de la estructura de la figura,



Para su resolución será necesario aplicar el equilibrio de fuerzas verticales y horizontales en todos los nudos donde no haya reacciones. Puesto que la estructura es estáticamente indeterminada es necesario considerar las fuerzas de tracción y compresión en cada barra, lo que conduce al sistema $ASA^T x = P$, donde P son las fuerzas aplicadas en los nudos, A la matriz que relaciona las fuerzas nodales con las cargas en ambas direcciones, S es una matriz diagonal que caracteriza las barras de la estructura (valores de aE/L , relacionando el área transversal de la barra con el módulo de Young y su longitud) y x los desplazamientos de los nudos. Posteriormente se obtienen las tensiones como $f = SA^T x$. En el ejemplo propuesto $S=(4255,6000,6000,3670,3000,3670,6000,6000,4255,3000)$.

Obtener la tensión octaédrica máxima y el máximo desplazamiento al aplicar los siguientes vectores de carga:

Carga debida a la nieve: $P_1=(0,-1000,0,0,0,-1000,0,0,0)$

Carga debida al paso de vehículos: $P_2=(0,0,100,-700,0,0,100,-700,100)$

Carga debida al viento lateral: $P_3=(0,0,0,0,-500,0,0,0,-500)$

Determinar las tensiones máximas y desplazamientos máximos así como las barras y nudos en que se producen. Repetir el proceso cuando se aplica una combinación lineal de las cargas anteriores, esto es, $P=c_1 P_1 + c_2 P_2 + c_3 P_3$ donde $c_1 + c_2 + c_3 = 1$ y $c_1, c_2, c_3 \in [0,1]$

Nudo	Equilibrio Eje X	Equilibrio Eje Y
1	$P_{1x} + R_{1x} - F_1 \cos \alpha - F_2 = 0$	$P_{1y} + R_{1y} - F_1 \sin \alpha = 0$
2	$P_{2x} + F_1 \cos \alpha - F_4 - F_6 \cos \beta = 0$	$P_{2y} + F_1 \sin \alpha + F_3 + F_6 \sin \beta = 0$
3	$P_{3x} + F_2 - F_5 \cos \beta - F_7 = 0$	$P_{3y} - F_3 - F_5 \sin \beta = 0$
4	$P_{4x} - F_4 + F_5 \cos \beta - F_9 \cos \alpha = 0$	$P_{4y} + F_5 \sin \beta + F_8 + F_9 \sin \alpha = 0$
5	$P_{5x} + F_6 \cos \beta + F_7 - F_{10} = 0$	$P_{5y} - F_6 \sin \beta - F_8 = 0$
6	$P_{6x} + R_{6x} + F_9 \cos \alpha + F_{10} = 0$	$R_{6y} + P_{6y} - F_9 \sin \alpha = 0$

Nota: $\alpha=45^\circ=\pi/4$, $\beta=30^\circ=\pi/6$

2 Anexo: Instrucciones necesarias

2.1 Operaciones con matrices

Función	Operación	Función	Operación
A'	Matriz transpuesta de A	<code>transpose(A)</code>	Matriz transpuesta de A
<code>inv(A)</code>	Matriz inversa de la matriz cuadrada A	<code>rank(A)</code>	Rango de la matriz A
<code>det(A)</code>	Determinante de la matriz cuadrada A	<code>length(v)</code>	Devuelve la longitud del vector v
<code>trace(A)</code>	Suma de los elementos de la diagonal de A	<code>size(A)</code>	Devuelve el orden (tamaño) de la matriz A

2.2 Generar tipos especiales de matrices

Función	Operación	Función	Operación
<code>diag(A)</code>	Extraer la diagonal de la matriz A	<code>diag(v)</code>	Matriz diagonal con los elementos de v
<code>tril(A)</code>	Parte triangular inferior de la matriz A	<code>triu(A)</code>	Parte triangular superior de la matriz A
<code>eye(n)</code>	Crea la matriz identidad de orden n-	<code>eye(m,n)</code>	Idem orden mxn.
<code>zeros(n)</code>	Crea la matriz nula de orden n	<code>zeros(m,n)</code>	Idem de orden mxn
<code>ones(n)</code>	Crea la matriz de unos de orden n	<code>ones(m,n)</code>	Idem de orden mxn
<code>rand(n)</code>	Crea una matriz aleatoria uniforme de orden n	<code>rand(m,n)</code>	Idem de orden mxn

2.3 Operaciones Lógicas con Matrices

`any(V)` Devuelve 0 si todos los elementos del vector V son nulos, y devuelve 1 si algún elemento de V es no nulo

`all(V)` Devuelve 1 si todos los elementos del vector V son no nulos, y devuelve 0 si algún elemento de V es nulo

`find(V)` Devuelve los lugares (o índices) que ocupan los elementos no nulos del vector V

2.4 Resolución de Sistemas y factorización

Función	Operación	Función	Operación
$X=A \setminus B$	Resuelve el sistema $A \cdot X=B$	$X=A/B$	Resuelve el sistema $X \cdot A=B$
<code>lu(A)</code>	Factorización LU	<code>chol(A)</code>	Factorización de Choleski

2.5 Autovalores y Autovectores

Función	Operación	Función	Operación
<code>eig(A)</code>	Autovalores de la matriz A	<code>[V,D]=eig(A)</code>	M matriz diagonal de autovalores D y matriz V de autovectores por columnas
<code>poly(A)</code>	Polinomio característico de A	<code>poly(V)</code>	Vector cuyas raíces son los elementos del vector V