Tema 9: Otros temas de aplicación

1. Introducción

Existen muchos elementos interesantes y aplicaciones del Matlab que no se han comentado a lo largo de los temas. Se invita al lector a que investigue sobre ellos según las líneas que sean más afines a su entorno de trabajo o estudio. No obstante, se quiere terminar este texto con un capítulo dedicado a introducir una variedad de conceptos más o menos usuales que no se han incluido anteriormente.

2. Polinomios

El programa dispone de comandos específicos para realizar las operaciones más comunes con polinomios, tal es el caso de búsqueda de raíces, evaluación en determinados valores, diferenciación, interpolación y ajuste.

Destacar que los polinomios en Matlab se introducen a partir de vectores cuyos elementos son los coeficientes del mismo. Si alguno no aparece se introduce como 0. Así, x^3+3x^2-5 será [1,3,0,-5].

Los comandos son:

- **polyval(u,x)**, evalúa el polinomio de coeficientes incluidos en el vector u en el valor indicado en x.
- **conv(u,v)**, da los coeficientes del polinomio resultado de multiplicar los polinomios de coeficientes incluidos en los vectores u y v.
- **[p,q]=deconv(u,v)**, devuelve los polinomios cociente y resto de la división entre los polinomios u y v.
- **roots(u)**, calcula las raíces del polinomio u
- **polyder(u)**, nos da el polinomio resultado de derivar u.
- **polyfit(x,y,n)**, polinomio de grado n que ajusta los puntos (x,y) en el sentido mínimos cuadrados.

- **poly(v)**, crea un polinomio cuyas raíces son las indicadas en el vector v.

3. Ajuste de datos. Interpolación

Además de la búsqueda de un polinomio interpolador, Matlab permite la interpolación a través de un gran número de técnicas. Destacaremos algunos de los comandos que realizan este tipo de aplicaciones:

- yi=**interp1(x,y,xi)**, da como resultado un vector yi tal que (xi,yi) es el conjunto total de puntos hallados por interpolación unidimensional del conjunto de puntos (x,y).
- yi=**interp1(x,y,xi,método)**, realiza interpolación mediante el método elegido (nearlest, lineal, cubic, v5cubic, spline o pchip).
- zi=**interp2(x,y,z,xi,yi)**, da como resultado un vector zi tal que (xi,yi,zi) es el conjunto total de puntos hallados por interpolación bidimensional del conjunto de puntos (x,y,z).
- yi=**spline(x,y,xi)**, da como resultado un vector yi tal que (xi,yi) es el conjunto total de puntos hallados por interpolación cúbica spline del conjunto de puntos (x,y).

4. Matrices dispersas

Existen trabajos, especialmente en ingeniería, donde es necesario utilizar matrices de gran tamaño pero con un número importante de ceros en su interior (matrices dispersas). Operar con este tipo de matrices a través de métodos convencionales puede implicar tiempos muy grandes para el cálculo. Matlab dispone de funciones para trabajar con estas matrices dispersas que ahorra tiempos de ejecución.

El programa almacena estas matrices dispersas guardando en memoria solamente los elementos no nulos junto con la posición que ocupan en la matriz. Así, utiliza tres elementos, los valores de las filas de elementos no nulos, los valores de las columnas de estos elementos y el valor que tienen.

Ejemplo: La matriz $A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & -3 & 0 \\ 0 & 0 & 0 & 9 \\ 2 & 0 & 0 & 0 \end{bmatrix}$ se introduciría como:

```
>> A=sparse([1,2,2,3,4],[1,2,3,4,1],[1,2,-3,9,2])
A =
   (1,1)        1
   (4,1)        2
```

```
(2,2)        2
(2,3)       -3
(3,4)        9
```

Para esto puede ser de utilidad el comando find, [i,j,v]=find(A) tiene como salida el vector de las filas y el de las columnas de los elementos no nulos de la matriz A así como el valor de dichos elementos.

De igual forma, el programa permite convertir una matriz llena en dispersa a través de este comando:

```
>> B=[1 2 0 0 0 0;0,9,0,0,8,0;0,0,0,-2,0,1]
B =
     1     2     0     0     0     0
     0     9     0     0     8     0
     0     0     0    -2     0     1
>> S=sparse(B)
S =
   (1,1)        1
   (1,2)        2
   (2,2)        9
   (3,4)       -2
   (2,5)        8
   (3,6)        1
```

El comando full realiza la operación contraria, el llenado de la matriz dispersa

```
>>full(S)
ans =
     1     2     0     0     0     0
     0     9     0     0     8     0
     0     0     0    -2     0     1
```

En el menú de ayuda matlab\sparfun - Sparse matrices se encuentran los comandos para trabajar con este tipo de matrices. El criterio general para trabajar con matrices dispersas en Matlab es que casi todas las operaciones matriciales estándar funcionan sobre ellas al igual que lo hacen sobre las llenas.

5. Álgebra lineal

Se destacan algunos temas relacionados con álgebra que pueden ser de interés y que se realizan con el programa a través de comandos específicos.

Valores propios:

El trabajo con valores y vectores propios es esencial en numerosas disciplinas. Matlab permite trabajar con esta materia con comandos entre los que destacamos:

eig(A): Halla los autovalores de la matriz cuadrada A.

[P,D]=eig(A): Determina la matriz diagonal D de los autovalores de A y la matriz P de columnas los autovectores correspondientes de forma que AP=PD.

Jordan(A): Halla la matriz canónica de Jordan de la matriz A.

[P,J]= Jordan(A): Halla la matriz canónica de Jordan de la matriz A y la matriz de paso P de forma que $P^{-1}AP=J$.

poly(A): Devuelve el polinomio característico de la matriz A

Descomposición de matrices:

Matlab trabaja con métodos de descomposición matricial como el LU, Cholesky, qr,...

[L,U]=lu(A): Descompone la matriz A en el producto A=LU, siendo L una matriz triangular inferior y U una superior.

[L,U,P]=lu(A): Da una matriz triangular inferior L, una superior U y una de permutación P de forma que PA=LU.

R=chol(A): Devuelve la matriz triangular superior R tal que R'R=A siempre que A sea definida positiva. En caso contrario devuelve un error.

[Q,R]=qr(A): Devuelve la matriz triangular superior R de igual dimensión que A y la matriz ortogonal Q de forma que A=QR.(Puede aplicarse a matrices no cuadradas)

[Q,R,E]=qr(A): devuelve la matriz triangular superior R de igual dimensión que A, la matriz ortogonal Q y la matriz de permutaciones E de forma que de forma que AE=QR.

Resolución de ecuaciones:

Matlab permite resolver ecuaciones. Algunos de los comandos para realizarlo son:

solve('ecuación','x'): Resuelve la ecuación en la variable x.

solve('ecuación1,eciación2,...ecuaciónn','x1,x2,...xn'): Resuelve el sistema de ecuación en las variables x1,...xn.

x=fzero(función,x0): Halla un cero de la función ceca de x0.

[x,feval]=fzero(función,x0): Da también el valor de f en x.

Nota: Existen numerosos comandos que intentan resolver ecuaciones y sistemas según diversos métodos numéricos.

Práctica 9: Otros temas de aplicación

1. Determinar las raíces del polinomio $x^3-6x^2-72x-27$, evaluarlo en alguna de ellas para verificar que lo son.
2. Introducir el polinomio $2x^3+4x^2+1$, se pide:
   a. Calcular sus raíces.
   b. Evaluarlo en x=3.
   c. Crear un polinomio de raíces –1,2,5.
   d. Multiplicar ambos polinomios.
3. Determinar el cociente y el resto de dividir los polinomios $5x^5+3x^3-x^2+x-1$ y $2x^2-3x+2$. Derivar el primero.
4. Construir una función que sume polinomios de cualquier grado y utilizarla para sumar los polinomios del ejercicio anterior.
5. Determinar el polinomio interpolador de segundo grado que pasa por los puntos (-1,4), (0,2), (1,6). Dibujar los puntos y el polinomio en el intervalo [-2,2].
6. Determinar el polinomio de ajuste de grado 1 en el sentido mínimos cuadrados para los puntos (1,2), (-2,7), (-1,6). Dibuja los puntos y el polinomio en un intervalo adecuado.
7. Los directores de una empresa se reúnen para analizar la situación financiera de la misma. Al estudiar la tabla:

AÑO	Beneficios
1	1
2	4
3	8
4	15
5	25

   prevén que la curva que mejor representa los beneficios durante los próximos años es un polinomio de segundo grado. Determinar los beneficios que esperan obtener el próximo año.
8. Hallar y representar 30 puntos de interpolación (x,y) de la función sen(x) para valores de x igualmente espaciados entre 0 y 10.
9. Repetir el problema 8 para interpolación spline y comparar las gráficas obtenidas.
10. Se considera un conjunto de temperaturas medidas sobre las cabezas de los cilindros de un motor para utilizar en coches de carreras. Los tiempos de

funcionamiento del motor en segundos y las temperaturas en grados Fahrenheit son las siguientes:

Tiempo	0	1	2	3	4	5
Temperatura	0	20	60	68	77	110

Realizar una regresión lineal que ajuste la temperatura en función del tiempo. Repetirlo para regresiones polinómicas de grados 2, 3 y 4, representando los resultados.

11. Definir las siguientes matrices de forma que sólo se guarden los elementos no nulos:

$$A=\begin{pmatrix}1 & -4 & 0 & 0 & 0\\ 7 & 3 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0\\ 0 & 0 & -13 & 11 & 0\\ 0 & 0 & 2 & 7 & -4\end{pmatrix} \quad B=\begin{pmatrix}1 & -1 & 0 & 0 & 0\\ 2 & 3 & 0 & 0 & -8\\ 0 & 0 & -2 & 3 & 0\\ 0 & 0 & 0 & -1 & 0\\ 0 & 0 & 2 & 7 & -4\end{pmatrix}$$

Se pide:

a. Recuperar la matriz A con todos sus elementos
b. Calcular A+B, A*B
c. Determinar los elementos no nulos de A*B junto con la posición que ocupan.

12. Dada la matriz $A=\begin{pmatrix}1 & -1 & 3\\ -1 & i & -1-2i\\ i & 1 & i-2\end{pmatrix}$ se pide:

a. Sus autovalores.
b. Sus autovectores.
c. Su polinomio característico.

13. Sea la matriz $A=\begin{pmatrix}1 & 5 & -2\\ -7 & 3 & 1\\ 2 & 2 & -2\end{pmatrix}$, se pide su descomposición LU, QR y Cholesky comprobando los resultados.

14. Resolver las ecuaciones:

a. xsen(x)=1/2
b. $x^4=1$

15. Resolver el sistema de dos ecuaciones dado por: $\begin{array}{l}\cos(\frac{x}{12})e^{\frac{x^2}{16}} = y\\ \frac{-5}{4}+y=sen(x^{\frac{3}{2}})\end{array}$