# Lección E

# Sistemas de ecuaciones lineales. Métodos directos

En esta lección estudiamos la solución de un sistema de Cramer $Ax = B$, lo que significa que $A$ es regular o invertible o que verifica $\det(A) \neq 0$, mediante un método directo. Siendo éste cualesquiera que permita, en ausencia de errores, mediante un número finito de pasos obtener la solución exacta. En propiedad, esto no ocurre en general debido a los inevitables errores de redondeo.

## E.1. Sistema triangular inferior. Sustitución progresiva

Consideremos el sistema $Lx = B$ con $L$ una matriz triangular inferior, es decir que verifica $l_{ij} = 0$ para $j > i$ lo que significa que nuestro sistema toma la forma:

$$\begin{array}{ccccccccc}
l_{11}x_1 & & & & & & & = & b_1 \\
l_{21}x_1 & + & l_{22}x_2 & & & & & = & b_2 \\
\vdots & & \vdots & & \ddots & & & & \vdots \\
l_{n1}x_1 & + & l_{n2}x_2 & + & \dots & + & l_{nn}x_n & = & b_n
\end{array}$$

La regla de sustitución progresiva consiste en despejar $x_1$ de la primera ecuación (es inmediato poniendo $x_1 = b_1/l_{11}$) y continuar el proceso mediante la sustitución de $x_1$ en la segunda ecuación para obtener el valor de $x_2$ y así sucesivamente. En definitiva, la única solución $(x_1, x_2, \dots, x_n)$ está caracterizada por ser:

$$x_k = \frac{1}{l_{kk}}\, b_k - \sum_{j=1}^{k-1} x_j\, \frac{l_{kj}}{l_{kk}}$$

**Práctica a** Con la orden `type e_sp` obtenemos el listado de la función `Mt\e_sp.m` definida en el libro [6], pág 163, y que permite resolver un sistema cuadrado, regular y triangular inferior en el que se dan como datos la matriz regular $L$ y el vector $B$. Como ejemplo de aplicación de dicha función resolvemos el sistema de ecuaciones

$$\begin{aligned}
x_1 &= 1 \\
2x_1 + 3x_2 &= 2 \\
x_1 + 2x_2 + 3x_3 &= 4
\end{aligned}$$

para lo que ejecutamos el guión

```
flops(0);            %comenzamos a contar el número de operaciones
A=[1 0 0;2 3 0;1 2 3]; b=[1 2 4].';
x=e_sp(A,b).',  flops
```

con él obtenemos la única solución del sistema, $x = (1, 0, 1)$, y el número de operaciones "flops" necesarias para obtenerlo, 12. En general, para un sistema regular y triangular inferior de $n$ ecuaciones reales el número de operaciones necesarias para resolverlo es del orden de $n(n+1) = O(n^2)$.

Resolvemos también los sistemas triangulares inferiores siguientes:

$$\begin{array}{lcl} 7x_1 & = & 7 \\ x_1 - x_2 & = & 4.1 \\ 0.5x_1 + 7.82x_2 - 24.97x_3 & = & -151.114 \end{array} \quad ; \quad \begin{array}{lcl} 28x_1 & = & 7 \\ 34x_1 - 26.2x_2 & = & 4.1 \\ 3x_1 + 7.82x_2 - 2.497x_3 & = & -15.1 \end{array}$$

## E.2. Sistema triangular superior. Sustitución regresiva

Consideramos ahora el sistema $Ux = B$ con $U$ una matriz triangular superior, es decir que verifica $s_{ij} = 0$ si $i > j$. La regla de sustitución regresiva consiste en despejar de abajo hacia arriba de forma análoga a la anterior con lo que obtenemos que la única solución $x = (x_1, \ldots, x_n)$ del sistema $Ux = B$ está dado por la regla:

$$x_k = \frac{1}{u_{kk}} b_k - \sum_{j=k+1}^{n} x_j \frac{u_{kj}}{u_{kk}}$$

**Práctica b** Con la orden `type e_sr` obtener y estudiar el listado de la función `Mt\e_sr.m` definida en el libro [6], pág 165, que resuelve mediante la regla de sustitución regresiva un sistema cuadrado, regular y triangular superior, a partir de los datos: $U$ la matriz del sistema y $B$ el vector de los términos independientes. Probar la eficacia del programa resolviendo los sistemas:

$$\begin{array}{rcl} x_1 + 4x_2 + 5x_3 & = & 1 \\ 3x_2 + 2x_3 & = & 2 \\ 3x_3 & = & 6 \end{array} \; ; \quad \begin{array}{rcl} 3x_1 - x_2 + 2x_3 & = & 12 \\ 7x_2 + 7x_3 & = & 21 \\ -21x_3 & = & -42 \end{array}$$

## E.3. Métodos exactos para sistemas generales

Antes de comenzar a describir los métodos de resolución de sistemas generales vamos a recordar algunos elementos de Álgebra Lineal Elemental

Sea $E\colon M_{m\times n}(K) \to M_{m\times n}(K)$ una transformación en el conjunto de matrices de orden $m \times n$. Diremos que $E$ es una *transformación elemental* si

- $E(A)$ se obtiene a partir de $A$ intercambiando la columna (*resp.* fila) $i$-ésima por la $j$-ésima, $i \neq j$.

- $E(A)$ se obtiene a partir de $A$ multiplicando la columna (*resp.* fila) $i$-ésima por $\lambda \neq 0$.
- $E(A)$ se obtiene a partir de $A$ sumando a la $i$-ésima columna (*resp.* fila) la $j$-ésima columna (*resp.* fila) multiplicada por $\lambda \neq 0$, $i \neq j$.

Es importante observar:

1. Si denotamos por $C(A)$ el resultado de una transformación elemental en columnas entonces

$$\boxed{C(A) = A \cdot C(\mathrm{Id}_n)}$$

   y que si $F(A)$ denota el resultado de una transformación en filas entonces

$$\boxed{F(A) = F(\mathrm{Id}_m) \cdot A}$$

   Además si $C = C_n \circ \cdots \circ C_1$ y $F = F_r \circ \cdots \circ F_1$ entonces también $C(A) = M \cdot C_n(\ldots(C_1(\mathrm{Id}_n))\ldots)$ y $F(A) = F_r(\ldots(F_1(\mathrm{Id}_m))\ldots) \cdot M$.

   Las dos igualdades encuadradas anteriores justifican que las matrices $C(\mathrm{Id}_n)$ y $F(\mathrm{Id}_m)$ se denominen matrices de las transformaciones elementales $C$ y $F$ respectivamente.

2. Cada transformación elemental posee determinante distinto de cero y su inversa es otra transformación elemental. En consecuencia, si $E$ es una transformación cualesquiera en filas o columnas y $F$ es una transformación en filas entonces se verifica:

   *a*) $\mathrm{Rango}\, E(A) = \mathrm{Rango}\, A$.

   *b*) Si $A^* = [A, B]$ es la matriz ampliada del sistema $Ax = B$ entonces $F(A^*)$ es la matriz ampliada de un sistema equivalente, en el sentido de que ambos poseen el mismo conjunto de soluciones.

Como consecuencia del apartado (b) anterior se deduce:

1. Todo sistema $Ax = B$ regular es equivalente a un sistema $Ux = B'$ donde $U$ es una matriz triangular superior.

2. Existe una permutación en columnas $P$ de la matriz regular $A$ de modo que $A \cdot P$ factoriza en el producto de una matriz triangular inferior $L$ y otra triangular superior $U$. Es decir, $LU = AP$.

   El anterior enunciado se puede también poner: Existe una permutación en filas $P$ de la matriz regular $A$ de modo que $P \cdot A$ factoriza en el producto de una matriz triangular inferior $L$ y otra triangular superior $U$. Es decir, $LU = PA$.

En efecto, la mecánica que convierte $A^*$ en una matriz triangular superior equivalente $U^*$ se puede detallar con un ejemplo:

Consideramos la matriz

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & b_1 \\ a_{21} & a_{22} & a_{23} & b_2 \\ a_{31} & a_{32} & a_{33} & b_1 \end{pmatrix}$$

Supuesto $a_{11} \neq 0$ podemos considerar la transformación elemental $F_1$ que en filas suma a la segunda el resultado de multiplicar por $-a_{21}/a_{11}$ la primera y que a la tercera fila suma el resultado de multiplicar por $-a_{31}/a_{11}$ la primera fila. Obtenemos:

$$F_1(A^*) = \begin{pmatrix} a_{11} & a_{12} & a_{13} & b_1 \\ 0 & a'_{22} & a'_{23} & b'_2 \\ 0 & a'_{32} & a'_{33} & b'_1 \end{pmatrix}; \qquad F_1 = \begin{pmatrix} 1 & 0 & 0 \\ -a_{21}/a_{11} & 1 & 0 \\ -a_{31}/a_{11} & 0 & 1 \end{pmatrix}$$

El proceso realizado corresponde a lo que se denomina utilizar $a_{11}$ como pivote. En el siguiente paso, supuesto $a_{22} \neq 0$, utilizando $a_{22}$ como pivote conseguimos:

$$F_2(A^*) = \begin{pmatrix} a_{11} & a_{12} & a_{13} & b_1 \\ 0 & a'_{22} & a'_{23} & b'_2 \\ 0 & 0 & a''_{33} & b''_1 \end{pmatrix}; \qquad F_2 = \begin{pmatrix} 1 & 0 & 0 \\ a & 1 & 0 \\ b & c & 1 \end{pmatrix}$$

Con lo que habríamos terminado ya que $F_2 \cdot A^* = U^*$ y también $F_2 \cdot A = U$ de donde $Ax = B$ es equivalente a un sistema triangular superior y $A = F_2^{-1}U = LU$. Lo descrito anteriormente puede fallar si alguno de los pivotes $a_{ii}$ que hemos ido eligiendo es 0. En dicho caso existe un elemento en su misma fila distinto de cero, $a_{ik} \neq 0$ para algún $k > i$, ya que en caso contrario la matriz $A$ sería singular, con lo cual si hacemos sobre $A$ la transformación elemental $T_i^k$ que corresponde a intercambiar las columnas $i, k$ tendríamos que a la matriz $A \cdot T_i^j(\mathrm{Id})$ se le puede aplicar un paso más en el proceso de factorizar como producto de dos matrices triangulares y en definitiva eso nos prueba que existe una permutación $P$ de modo que $A \cdot P$ es factorizable de la forma $A \cdot P = LU$ con $L$ una matriz triangular inferior y $U$ una matriz triangular superior. Si aplicamos esto a la matriz $A^t$ concluimos que existe $P = P^t$ de modo que $L'U' = A^tP^t$ pero entonces $U'^tL'^t = PA$ luego $LU = PA$.

Describimos ahora los algoritmos más importantes de resolución directa de sistemas regulares generales.

**Eliminación gaussiana**

La eliminación gaussiana es la que empleamos para reducir un sistema regular $Ax = B$ a un sistema triangular superior $Ux = B'$.

El coste operativo de resolver un sistema utilizando la eliminación gaussiana es $\mathrm{O}(\frac{2}{3}n^3)$.

**Factorización LU**

Sea $P$ una permutación en filas de la matriz $A$ para la que existe la factorización $P \cdot A = LU$, claramente para $P$ se verifica $P^t = P^{-1}$. El sistema $Ax = B$ se puede expresar $LUx = P \cdot B$ de modo que se puede reducir a la resolución de los sistemas:

$$\begin{array}{ll} Lz = P \cdot B & \text{Triangular inferior} \\ Ux = z & \text{Triangular superior} \end{array}$$

La factorización LU está implementada en Matlab mediante la orden

```
[L,U,P]=lu(A)
```

y donde $P \cdot A = LU$.

El coste operativo de resolver un sistema utilizando la factorización LU es el mismo que en la eliminación gaussiana pero tiene la ventaja de que una vez conseguida la factorización $P \cdot A = LU$ la podemos emplear en todos aquellos sistemas $Ax = B$ que tengan la misma matriz $A$. Un ejemplo de esta situación es el cálculo de la inversa.

### Factorización de Cholesky

Tal como se puede ver en el último problema de Álgebra Lineal de primero toda matriz simétrica $A$ definida positiva posee una única descomposición $A = L^t L$ con $L$ una matriz triangular inferior. En consecuencia, el sistema $Ax = B$ es equivalente a resolver:

$$\begin{array}{ll} L^t z = B & \text{Triangular inferior} \\ Lx = z & \text{Triangular superior} \end{array}$$

La factorización de Cholesky está implementada en Matlab mediante la orden

```
L=chol(A)
```

y donde $A = L^t L$.

El coste operativo de la resolución por el método de Cholesky es $= O(\frac{1}{3}n^3)$.

### Factorización QR

Se factoriza $A$ en la forma $A = QR$, con $Q$ una matriz ortogonal, $Q^t Q = \mathrm{Id}$, y $R$ triangular superior. De este modo el sistema $Ax = B$ queda equivalente a la resolución de los sistemas:

$$\begin{array}{ll} Qz = B & \text{De solución } z = Q^t \cdot B \\ Rx = z & \text{Triangular superior} \end{array}$$

La factorización QR está implementada en Matlab mediante la orden

```
[Q,R]=qr(A)
```

y donde $A = QR$.

El coste operativo de la factorización QR es $\mathrm{O}(2n^3)$.

### Método empleado por Matlab

El método que emplea Matlab para resolver el sistema regular $Ax = B$ cuando ejecutamos la orden

```
A\B
```

es el óptimo de entre los anteriores. Con la precaución de que no siempre Matlab resuelve el sistema $AX = B$. Esto que decimos lo observamos en las siguientes prácticas:

**Práctica c** Consideramos el sistema $x+2y = 3$ del cual observamos que posee infinitas soluciones. Si ejecutamos en Matlab el listado

```
A=[1 2], b=[3]
x=A\b, x=x.'
```

obtenemos como respuesta $x = (0, 1.5)$ de lo que podemos observar que Matlab sólo nos da una de las soluciones.

**Práctica d** Consideramos el sistema

$$\left.\begin{array}{rcrcl} r_1 & : & x - y & = & 0 \\ r_2 & : & x + y & = & 0 \\ r_3 & : & 2x + y & = & 1 \end{array}\right\} \tag{E.1}$$

Con el listado

```
x=-0.5:0.05:1.5; y1=x; y2=-x; y3=1-2*x;
a1= 0.2857;   a2= 0.1429;
plot(x,y1,'r',x,y2,'b',x,y3,'g',a1,a2,'+')
```

obtenemos la representación gráfica de las rectas $r_1$, $r_2$ y $r_3$ que definen el sistema (E.1). De esta representación observamos que el sistema es incompatible ya que no hay intersección común a las tres rectas. Sin embargo, si resolvemos el sistema por Matlab, es decir, si ejecutamos

```
A=[1 -1; 1 1; 2 1], b=[0 0 1].'
a=A\b;
```

obtenemos como única solución $a =$ (`a1,a2`) $= (0.2857; 0.1429)$. ¿Qué significa esto?:

Del sistema $AX = b$ dado en (E.1) obtenemos $A^tAx = A^tb$ multiplicando ambos miembros por la traspuesta de $A$. Para este nuevo sistema se verifica que su matriz asociada, $A^tA$, es regular. En efecto, $A^tA$ es claramente cuadrada y verifica que $\operatorname{Ker} A^tA = \operatorname{Ker} A = 0$.

> La igualdad $\operatorname{Ker} A^tA = \operatorname{Ker} A$ es general y fácil de probar: Claramente $\operatorname{Ker} A^tA \supset \operatorname{Ker} A$ y ahora si $x \in \operatorname{Ker} A^tA$ entonces $x^t(A^tAx) = 0$ lo que implica que $||Ax||_2 = \sqrt{x^tA^t \cdot Ax} = 0$ y por tanto que $Ax = 0$ y $x \in \operatorname{Ker} A$.

En consecuencia, $A^tAx = A^tb$ posee una única solución de la que podemos comprobar con el listado `c=(A.'*A)\(A.'*b)` que coincide con $a$.

Además, para todo vector $z$ se tiene $z^t(A^tAc - A^tb) = 0$ y por tanto que $A(z)$ es perpendicular a $Ac - b$ para todo $z$ de donde

$$||Ay - b||_2 = ||Ac - b + A(y - c)||_2 = ||Ac - b||_2 + ||A(y - c)||_2 \tag{E.2}$$

lo que implica que la expresión $||Ay - b||_2$ es mínima precisamente cuando $y = c$.

Podemos resumir todo lo anterior diciendo: Para $A$ una matriz inyectiva, es decir, tal que su rango sea máximo, se verifica que es equivalente:

- Resolver el sistema regular $A^tAx = A^tb$.
- Resolver en el sentido de los *mínimos cuadrados* el sistema $Ax = b$. Por ello entendemos obtener el vector $c$ con la propiedad de ser $||Ac - b||_2$ mínimo.

- Calcular $c = (A^tA)^{-1}A^tb$. La aplicación cuya matriz es $(A^tA)^{-1}A^t$ recibe el nombre de *pseudo-inversa* de $A$ y está implementada en Matlab con el comando `pinv(A)`.
- Ejecutar `A\b` ó `pinv(A)*b` en Matlab.

Como observación final a esta práctica y a esta exposición decir que si la transformación $T_A\colon E \to F$ no es inyectiva entonces la transformación que $T_A$ induce en el cociente de forma natural, $\overline{T}_A\colon E/\operatorname{Ker} T_A \to F$, es inyectiva con lo que todo lo anterior es directamente aplicable a $\overline{T}_A$ y por tanto podemos afirmar:

- El sistema $A^tAx = A^tb$ es siempre compatible y su conjunto de soluciones es una variedad afín de dirección $\operatorname{Ker} A$.
- El sistema $Ax = b$ siempre es compatible en el sentido de los mínimos cuadrados y su conjunto de soluciones es una variedad afín de dirección $\operatorname{Ker} A$.

**Práctica dd** Matlab también cuenta con la posibilidad de resolver de forma 'exacta' o simbólica un sistema de ecuaciones.

Como ejemplo resolvemos simbólicamente y numéricamente los siguientes sistemas lineales

$$\begin{bmatrix} 4 & -3 & 3 \\ 6 & 1 & -9 \\ 2 & -5 & -6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 9 \\ 5 \end{bmatrix}; \qquad \begin{bmatrix} 2 & 1 & 3 \\ 4 & 2 & -1 \\ 6 & 3 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

con el listado

```
A1=[4 -3 3;6 1 -9;2 -5 -6]; As1=sym(A1)
b1=[0 9 5].'
x1=A1\b1                %solución numérica
xs1=As1\b1              %solución simbólica

A2=[2 1 3;4 2 -1;6 3 2]; As2=sym(A2)
b2=[0 0 0].'
x2=A2\b2                %solución numérica
xs2=As2\b2              %solución simbólica
```

## E.4. Método directo para sistemas tridiagonales

Los sistemas tridiagonales aparecen en la resolución de numerosos problemas numéricos, de aquí su importancia.

Un sistema tridiagonal tiene la forma:

$$\begin{bmatrix} B_1 & C_1 & & & & & \\ A_2 & B_2 & C_2 & & & & \\ & A_3 & B_3 & C_3 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & A_i & B_i & C_i & \\ & & & & \ddots & \ddots & \ddots \\ & & & & & A_n & B_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_i \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ \vdots \\ d_i \\ \vdots \\ d_n \end{bmatrix}$$

Es fácil observar lo siguiente:

1. Que utilizando como pivotes los sucesivos elementos de la diagonal desde $i = 1$ hasta $i = n$ la matriz ampliada del sistema tridiagonal anterior es equivalente a la matriz:

$$\begin{bmatrix} B'_1 & C_1 & & & & & d'_1 \\ & B'_2 & C_2 & & & & d'_2 \\ & & B'_3 & C_3 & & & d'_3 \\ & & & \ddots & \ddots & & \vdots \\ & & & & B'_i & C_i & d'_i \\ & & & & & \ddots \quad \ddots & \vdots \\ & & & & & B'_n & d'_n \end{bmatrix}$$

donde los elementos $B'_i, d'_i$ y, por tanto, la solución es fácilmente calculable utilizando el siguiente algoritmo:

*a*) Definimos las variables $B'_i, d'_i$, $i = 1, \ldots, n$ por igualdades

$$\begin{aligned} B'_1 &= B_1; & d'_1 &= d_1; & & \\ B'_i &= B_i - rC_{i-1}; & d'_i &= d_i - rd'_{i-1}; & & i = 2, \ldots, n \end{aligned}$$

donde $r$ es la variable auxiliar $A_i/B'_{i-1}$.

*b*) Calculamos la última componente de la solución con

$$x_n = \frac{d'_n}{B'_n}$$

*c*) Calculamos las siguientes componentes en orden decreciente por la igualdad:

$$x_i = \frac{d'_i - C_i x_{i+1}}{B'_i}$$

para $i = n - 1, \ldots, 2, 1$.

2. Que lo anterior no es directamente aplicable si uno de los elementos diagonales, pongamos $B'_i$ es cero, en cuyo caso la ecuación $i$-ésima tomaría la forma $C_i x_{i+1} = d'_i$. Pero entonces, la

incógnita $x_{i+1}$ se obtendría de forma automática por la igualdad $x_{i+1} = d'_i/C_i$ con lo que el sistema quedaría reducido a un sistema de orden $n-1$ en las variables $\{x_1, \ldots, \widehat{x_{i+1}}, \ldots, x_n\}$ que es **tridiagonal** y por tanto resoluble mediante el anterior algoritmo o reducible a un sistema tridiagonal de orden estrictamente inferior y por tanto también resoluble mediante una ligera modificación del algoritmo descrito en el punto anterior (ver la práctica EG).

**Práctica e** Ejecutar el comando `type e_trid` con el fin de estudiar el listado de dicho archivo y comprobar que corresponde a la implementación del algoritmo descrito en el punto 1 anterior. Vamos ahora a utilizar la función citada para resolver el sistema tridiagonal

$$\begin{array}{rcrcrcrcl} 2.1x_1 & - & x_2 & & & & & = & 1 \\ x_1 & - & 2.2x_2 & + & x_3 & & & = & 1.6 \\ & & x_2 & - & 3.1x_3 & + & x_4 & = & 2.1 \\ & & & & x_3 & - & 2x_4 & = & 1.2 \end{array}$$

para lo que ejecutamos el siguiente guión:

```
b=[2.1 -2.2 -3.1 -2], a=[1 1 1], c=[-1 1 1], d=[1 1.6 2.1 1.2],
e_trid(a,b,c,d)
```

obteniendo como respuesta la solución $(-0.2927; -1.6146; -1.6595; -1.4297)$.

**Práctica f** Las matrices tridiagonales son un tipo particular de lo que se denominan matrices *huecas* o *dispersas*. Matlab cuenta con un comando especial para tratar dichas matrices, se trata del comando `sparse`. En esta práctica vemos su utilidad:

Consideramos una matriz tridiagonal con la diagonal principal idénticamente igual a 4, y las diagonales superior e inferior igual a 2, y estudiamos la variación en el número de operaciones a realizar al tratar la matriz como llena o como hueca:

```
D=sparse(4*diag(ones(1,32)));                   %Se define la diagonal,
DInf=sparse(2:32,1:31,2*ones(1,31),32,32);      %los elementos infradiagonales,
A=D+DInf+DInf.';                                %la matriz del sistema,
Allena=full(A);                                 %la matriz llena A
b=[1:32]';                                      %los términos independientes
flops(0), y=A\b; oper_hueca=flops, flops(0), x=Allena\b;
oper_llena=flops, norm(y-x)
```

Obtenemos, `oper_hueca=645`, `oper_llena=15400` y $\|y - x\| = 1.7381e - 14$ y observamos que para un sistema de orden 32 el número de operaciones se reduce en un 95 % respecto al sistema pleno.

Podemos también comparar el comando `sparse` con la función `e_trid` con el listado:

```
D=sparse(4*diag(ones(1,32)));
DInf=sparse(2:32,1:31,2*ones(1,31),32,32); A=D+DInf+DInf.';
b=[1:32]'; flops(0), y=A\b; oper_hueca=flops, a=2*ones(1,31);
diago=4*ones(1,32); flops(0), x=e_trid(a,diago,a,b);
oper_trid=flops, norm(y-x)
```

y comprobar que la función `e_trid` necesita aproximadamente la mitad de operaciones.

**Práctica puntuable g** (0.4 puntos) Modificar el fichero `e_trid` de la forma necesaria para que pueda resolver sistemas tridiagonales generales y no sólo aquellos en los que todos los elementos de la diagonal $B_i'$ son distintos de cero. La modificación que hay que introducir es la especificada en el punto 2 de la enumeración del comienzo de esta sección.

## E.5. Estabilidad

**Práctica h** Consideramos la matriz y los vectores siguientes (el ejemplo presente de mal condicionamiento o de inestabilidad es debido a R. S. Wilson):

$$A = \begin{bmatrix} 10 & 7 & 8 & 7 \\ 7 & 5 & 6 & 5 \\ 8 & 6 & 10 & 9 \\ 7 & 5 & 9 & 10 \end{bmatrix}; \quad b = \begin{bmatrix} 32 \\ 23 \\ 33 \\ 31 \end{bmatrix}; \quad c = \begin{bmatrix} 32.1 \\ 22.9 \\ 33.1 \\ 30.9 \end{bmatrix}$$

Con el listado

```
A=[10 7 8 7;7 5 6 5;8 6 10 9;7 5 9 10]; det(A) b=[32 23 33 31].';
c=[32.1 22.9 33.1 30.9].'; x=A\b, y=A\c,
```

obtenemos que $A$ es una matriz simétrica de determinante 1 para la que las soluciones a los sistema $Ax = b$, $Ay = c$ son respectivamente $x = (1, 1, 1, 1)$ y $y = (9.2; -12.6; 4.5; -1.1)$. Con ello observamos que a pesar de estar $b$ y $c$ separados únicamente por una décima en cada una de sus componentes se verifica que $x$ e $y$ están separados en 13.6 unidades en una de sus componentes lo que significa que los errores relativos se han multiplicado, con la norma infinito, por un factor del orden de ¡4300!. Con la norma 2 los errores relativos se han multiplicado por un factor 1352.

El condicionamiento de sistemas de ecuaciones lineales trata de explicar por qué se produce este fenómeno. Las relaciones

$$\|b\| = \|Ax\| \leq \|A\|\|x\| \tag{E.3}$$

$$\frac{\|x-y\|}{\|x\|} \leq \frac{\|A^{-1}b - A^{-1}c\|}{\|b\|/\|A\|} \leq \|A^{-1}\|\|A\|\frac{\|b-c\|}{\|b\|} \tag{E.4}$$

nos dicen que el parámetro que mide el factor de propagación de los errores relativos de la solución viene dado por el número

$$\mathrm{cond}(A) = \|A^{-1}\|\|A\|, \qquad \text{donde } \|\cdot\| \text{ es una norma inducida}$$

a dicho número se le denomina *número de condición* de la matriz invertible $A$.

Matlab permite el cálculo directo del número de condición, con la norma 2, de una matriz invertible $A$ con el comando `cond(A)`. En nuestro ejemplo resulta `cond(A)=2984`. Una estimación del número 1/`cond(A)` se consigue con el comando `rcond(A)`, este comando tiene la ventaja de que se computa con muchas menos operaciones.

La desigualdad (E.3) es una igualdad para los vectores $x$ y $b-c$ que verifican

$$\|A(x)\| = \|A\|\|x\|, \quad \|A^{-1}(b-c)\| = \|A^{-1}\|\|b-c\| \tag{E.5}$$

De ello se deduce que si $\|\cdot\|$ es una norma inducida entonces el número de condición es una cota para el refuerzo de la propagación de los errores relativos que **siempre se alcanza**. Es decir, siempre existen sistemas $Ax=b$, $Ax=c$ de modo que

$$\frac{\|x-y\|}{\|x\|} = \text{cond}(A)\cdot\frac{\|b-c\|}{\|b\|} \tag{E.6}$$

**Práctica i** Como ejemplo de nuestra última afirmación vamos a encontrar vectores $x$ y $b-c$ que verifican la igualdad (E.5) anterior. Ello equivale, claramente, a encontrar vectores $v_1 = x$ y $v_2 = b-c$ que sean máximos para las funciones $f(x) = \frac{\|Ax\|}{\|x\|}$ y $g(x) = \frac{\|A^{-1}(x)\|}{\|x\|}$ respectivamente. Denotemos entonces $M(A)$ al conjunto de máximos de la función $f$ y $m(A)$ a su conjunto de mínimos. Por ser $A$ una matriz simétrica es ortogonalmente diagonalizable, de hecho ejecutando

```
[V,D]=eig(A)
```

comprobamos que $A = VDV^{-1}$ para $D = \text{diag}(0.0102, 0.8431, 3.8581, 30.2887)$ y $V$ la matriz ortogonal dada por la igualdad

$$V = \begin{bmatrix} 0.5016 & -0.3017 & -0.6149 & 0.5286 \\ -0.8304 & 0.0933 & -0.3963 & 0.3803 \\ 0.2086 & 0.7603 & 0.2716 & 0.5520 \\ -0.1237 & -0.5676 & 0.6254 & 0.5209 \end{bmatrix}$$

La igualdad $A = VDV^{-1}$ con $V$ ortogonal implica:

1. $\|A\| = \|D\| = \text{máx}_i |d_{ii}|$, $\|A^{-1}\| = \|D^{-1}\| = \dfrac{1}{\text{mín}_i |d_{ii}|}$, para $D = (d_{ij})$.
2. $M(A) = V(M(D))$.
3. $M(A^{-1}) = V(m(D))$.
4. $\text{cond}(A) = \dfrac{\text{máx}_i |d_{ii}|}{\text{mín}_i |d_{ii}|}$

Ahora, es claro que $(0,0,0,1)^t \in M(D)$ y $(1,0,0,0)^t \in m(D)$ de modo que $v_1 = V(:,4) = (0.5286, 0.3803, 0.5520, 0.5209)^t$ y $v_2 = 10^{-1}V(:,1) = 10^{-1}(0.5016, -0.8304, 0.2086, -0.1237)^t$ son elementos de $M(A)$ y $M(A^{-1})$ respectivamente. Todo lo anterior lo podemos llevar al ordenador con el listado:

```
A=[10 7 8 7;7 5 6 5;8 6 10 9;7 5 9 10]
[V,D]=eig(A); x=V(:,4); v2=0.1*V(:,1);
b=A*x; c=b-v2; rb=norm(b-c)/norm(b); y=A\c;
rx=norm(x-y)/norm(x); factor=rx/rb; resultados=[x,y,b,c]
fprintf('....x....\t ....y....\t ....b....\t ....c....\t\n')
fprintf('%f\t %f\t %f\t  %f\t\n',resultados.'),
disp('factor de propagación y número de condición')
      factor, condic=cond(A)
```

Los vectores $b = (16.0096, 11.5176, 16.7180, 15.7781)^t$ y $c = (15.9595, 11.6007, 16.6971, 15.7905)^t$ tienen la propiedad de que las soluciones $x$ e $y$ de los sistemas $Ax = b$ y $Ay = c$ verifican la igualdad (E.6).

**Práctica j** Veamos ahora cómo se procede al cálculo de vectores $v_1 \in M(A)$ y $v_2 \in M(A^{-1})$ cuando $A$ es una matriz regular general que para fijar ideas supondremos que es la matriz de Vandermonde del vector $(1, 2, 3.5, 5)$.

Ahora $A$ no es simétrica pero en cualquier caso sí lo es $A^tA$; en consecuencia, $A^tA$ es ortogonalmente diagonalizable y de esto se deduce que existe una descomposición

$$A = UDV^t$$

con $D$ una matriz diagonal de valores no negativos y $U, V$ ortogonales (ver [3], pág. 264)

Se llama *descomposición en valores singulares* de la matriz $A$ a cualesquiera de las descomposiciones $A = UDV^t$ en que $D$ es una matriz diagonal y $U, V$ ortogonales.

Es importante observar:

- La matriz $D$ es única, salvo permutación en los elementos de la diagonal, ya que los elementos diagonales de $D^2$ coinciden con los valores propios de $A^tA$.
- Las afirmaciones 1–2) de la práctica anterior siguen siendo ciertas ahora para $A = UDV^t$ una descomposición en valores singulares. Además del hecho de ser $A^{-1} = VD^{-1}U^t$ una descomposición de $A^{-1}$ en valores singulares se deduce que la afirmación 3) de la práctica anterior es cierta a condición de poner $M(A^{-1}) = U(m(D))$.

Matlab cuenta con un comando, `svd`, que permite el cálculo de una descomposición en valores singulares. Lo aplicamos ahora al cálculo de los vectores $v_1, v_2$ con el listado:

```
A=vander([1 2 3.5 5]);    %A es la matriz de vandermonde de (1 2 3.5 5)
[U,D,V]=svd(A), x=V(:,1);
v2=0.1*U(:,4); b=A*x; c=b-v2;
rb=norm(b-c)/norm(b); y=A\c; rx=norm(x-y)/norm(x); factor=rx/rb;
resultados=[x,y,b,c] fprintf('....x....\t ....y....\t ....b....\t
....c....\t\n')
fprintf('%f\t %f\t %f\t  %f\t\n',resultados.'),
display('factor de propagación y número de condición')
      factor, condic=cond(A)
```

Los vectores $b = (1.2375, 8.7417, 44.5976, 127.5692)^t$ y $c = (1.1859, 8.8174, 44.5586, 127.5781)^t$ tienen la propiedad de que las soluciones $x$ e $y$ de los sistemas $Ax = b$ y $Ay =$ verifican la igualdad (E.6).

## E.6. Ejercicios

**Práctica p** Sean los números complejos $p = 4 + 1.5i$, $q = -2 - 0.5i$, $r = 6 + 2i$ y $s = 8 + 2.5i$. Se pide resolver el sistema

$$\begin{aligned} px_1 + qx_2 &= 5 \\ qx_1 + rx_2 + qx_3 &= 0 \\ qx_2 + rx_3 + qx_4 &= 0 \\ qx_3 + sx_4 &= 0 \end{aligned}$$

**Práctica q** Resolver los sistemas de ecuaciones siguientes:

$$1) \left\{ \begin{aligned} (-1+i)z_1 + (2-2i)z_2 + 4z_3 &= 1 \\ (2-2i)z_1 + 3z_2 + iz_3 &= 4 \\ 3z_1 + (1-5i)z_2 + (1+2i)z_3 &= -1 \end{aligned} \right. \; ; \quad 2) \left\{ \begin{aligned} x_1 + 2x_2 + 4x_3 &= 11 \\ 4x_1 - x_2 + x_3 &= 8 \\ 2x_1 + 5x_2 + 2x_3 &= 3 \end{aligned} \right. \; ;$$

$$3) \left\{ \begin{aligned} x_1 + 3x_2 + 2x_3 &= 0 \\ 2x_1 - x_3 &= 0 \\ 3x_1 + 3x_2 + x_3 &= 0 \end{aligned} \right. \; ; \quad 4) \left\{ \begin{aligned} x_1 + x_2 + x_4 &= 2 \\ 2x_1 + x_2 - x_3 + x_4 &= 1 \\ 4x_1 - x_2 - 3x_3 + 2x_4 &= 0 \\ 3x_1 + x_2 - x_3 + 7x_4 &= -3 \end{aligned} \right. \; ;$$

**Práctica r** Resolver el sistema tridiagonal

$$\begin{array}{rcrcrcrcrcr} x_1 & + & x_2 & & & & & & & = & 2 \\ -x_1 & + & x_2 & - & x_3 & & & & & = & -3 \\ & & 2x_2 & - & 4x_3 & + & 3x_4 & & & = & 2 \\ & & & & x_3 & - & x_4 & + & x_5 & = & 0 \\ & & & & & & x_4 & - & x_5 & = & 1 \end{array}$$

**Práctica s** Considerar el siguiente sistema

$$\left\{ \begin{aligned} x_2 + x_3 &= 6 \\ x_2 + 3x_3 - x_4 &= 1 \\ x_3 - x_5 &= 3 \\ x_4 + x_5 &= -1 \\ -5x_1 + x_6 &= 1 \\ x_1 + 2x_2 &= 0 \end{aligned} \right.$$

Se pide resolverlo como sistema hueco y pleno comparando en cada caso el número de operaciones

**Práctica puntuable t** (0.3 puntos) Consideramos el sistema de orden $n$ definido por la igualdad

$$\begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & -1 & 2 & -1 & \\ & & \ddots & \ddots & \ddots \\ & & & -1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$$

Se pide:

1. Comprobar que el vector $x = \frac{1}{n+1}(1, 2, \ldots, n)$ es la única solución del sistema anterior.
2. Resolver el sistema anterior para los casos $n = 10$, $n = 20$ y $n = 30$ utilizando el comando '\', el comando `sparse` y la función `e_trid`. Además, se pide comparar la eficacia entre ambos métodos obteniendo para ello una tabla donde se refleje la distancia de la solución exacta y el número de operaciones necesarias para obtener la solución en cada caso y para cada uno de los métodos.
3. Resolver el sistema anterior para los casos $n = 400$, $n = 600$ y $n = 800$ utilizando la función `e_trid`. Además, se pide calcular la distancia con la solución exacta y el número de operaciones necesarias para obtener la solución.

**Práctica u** Sea $H_{10}$ la matriz de Hilber de orden 10; se pide encontrar valores para $b$ y $c$ de modo que los sistemas $H_{10}x = b$ y $H_{10}y = c$ verifiquen:

1. $\|b - c\| < 0.05$.
2. $\frac{\|x-y\|}{\|x\|} = \operatorname{cond}(H_{10}) \cdot \frac{\|b-c\|}{\|b\|}$

**Práctica v** Para $n = 3$ hasta $n = 20$ se pide calcular el número de condición de la matriz de Hilbert $H_n$ de orden $n$ y representar en una gráfica el resultado obtenido.

**Práctica w** Para $n = 3, 7, 11, 15, 19, 23$ resolver el sistema de ecuaciones $H_n x = b$ para $b = H_n(1, 1, \ldots, 1)$ y $H_n$ la matriz de Hilbert. La solución debe ser el vector cuyas componentes son todas iguales a 1; sin embargo, comprobar cómo para valores pequeños de $n$ la solución en exacta y como se va perdiendo precisión para dimensiones más grandes.

(Apartado puntuable con 0.2 puntos) Obtener una gráfica que represente para cada $n = 1, \ldots, 20$ la distancia con la solución exacta.

**Práctica puntuable x** (0.4 puntos) Demostrar que una matriz tridiagonal con diagonal dominante (ver la práctica FU) puede escribirse como el producto de dos matrices del tipo (factorización de Crout):

$$L = \begin{pmatrix} l_{11} & 0 & \ldots & \ldots & 0 \\ l_{21} & l_{22} & 0 & \ldots & 0 \\ 0 & l_{32} & l_{33} & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & l_{nn-1} & l_{nn} \end{pmatrix} ; \quad U = \begin{pmatrix} u_{11} & u_{12} & 0 & \ldots & 0 \\ 0 & u_{22} & u_{23} & \ldots & 0 \\ 0 & & u_{33} & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & 0 & u_{nn} \end{pmatrix}$$