==

Práctica 5. Resolución de sistemas lineales: factorización QR, sistemas sobredeterminados y métodos iterativos

Contenidos

Factorización QR

Ejercicio 1. Factorización QR de Householder (ya estaba en la práctica anterior)

Escribir una función `[Q, R] = QRHouseholder(A)` para calcular la descomposición QR de la matriz del sistema de ecuaciones lineales $A = QR$ con R triangular superior por el método de Hausholder. (Si se quiere comparar, en MATLAB `qr()` proporciona una descomposición QR.) Usarla para resolver los siguientes sistemas de ecuaciones lineales $A\mathbf{x} = \mathbf{b}$ sin usar funciones de MATLAB que resuelvan sistemas de ecuaciones o inviertan matrices. Calcular el vector residuo y su norma 2.

$$a)\ \left\{\begin{array}{rcr} 6x_1 + 2x_2 - x_3 - x_4 & = & 0 \\ 2x_1 + 4x_2 + x_3 & = & 7 \\ -x_1 + x_2 + 4x_3 - x_4 & = & -1 \\ -x_1 - x_3 + 3x_4 & = & -2 \end{array}\right.$$

```
Q = 4×4
     -0.925820099772551        0.202804574998328       -0.158718373828798 ···
     -0.308606699924184       -0.878819824992753        0.286847388337863
      0.154303349962092       -0.412369302496599       -0.829952805693858
      0.154303349962092       -0.128442897498941        0.451337339396799
R = 4×4
      -6.48074069840786        -2.93176364927975         1.08012344973464 ···
  -3.88578058618805e-16        -3.52203945247096        -2.60265871247854
   1.66533453693773e-16                        0        -3.32558280000557
   2.22044604925031e-16     1.11022302462516e-16     1.11022302462516e-16

x = 4×1
      -1.62913907284768
       3.03311258278146
      -1.87417218543046
      -1.83443708609271

vec_residuo = 4×1
  -8.88178419700125e-16
```

```
       1.77635683940025e-15
       6.66133814775094e-16
      -4.44089209850063e-16
norm2_residuo =
       2.14132062325966e-15
```

$$b)\quad \begin{cases} 0.05x_1 + 0.07x_2 + 0.06x_3 + 0.05x_4 &= 0.23 \\ 0.07x_1 + 0.10x_2 + 0.08x_3 + 0.07x_4 &= 0.32 \\ 0.06x_1 + 0.08x_2 + 0.10x_3 + 0.09x_4 &= 0.33 \\ 0.05x_1 + 0.07x_2 + 0.09x_3 + 0.10x_4 &= 0.31 \end{cases}$$

```
Q = 4×4
         -0.430331482911935       -0.0843949472569712        -0.369528575478261 ···
         -0.602464076076709        -0.573885641347411        -0.257063356854442
         -0.516397779494322         0.810191493666933         -0.12853167842722
         -0.430331482911935       -0.0843949472569732         0.883655289187144
R = 4×4
         -0.116189500386223        -0.161804637574888        -0.164386626472359 ···
       8.67361737988404e-18      -0.00438853725736254        0.0224490559703546
       6.93889390390723e-18                         0        0.0239390251070699
      -6.93889390390723e-18     -1.56125112837913e-17     -6.93889390390723e-18

x = 4×1
           1.00000000000038
          0.999999999999772
          0.999999999999907
           1.00000000000006

vec_residuo = 4×1
      -8.32667268468867e-17
      -5.55111512312578e-17
                          0
      -5.55111512312578e-17
norm2_residuo =
       1.14439169963056e-16
```

Sistemas sobredeterminados

Ejercicio 2. Sistema sobredeterminado

Determinar la solución que minimice la norma del vector residuo para el siguiente sistema lineal:

$$S = \begin{cases} x + y + z = 2 \\ x + z + 2t = 3 \\ x + y + t = 4 \\ -y + 2z = 2 \\ -x + y - z + t = 1 \end{cases}$$

- Escribir A y $\mathbf{b}$ y calcular el rango de A.

```
rangoA =
     4
```

- Comprobar que el sistema de ecuaciones normales ($A^tA\mathbf{x} = A^t\mathbf{b}$) es compatible determinado.

```
compatible = logical
   1
```

- Calcular la solución $\mathbf{x}$ del sistema de ecuaciones normales.

```
x = 4×1
         0.142857142857143
          1.28571428571429
          1.28571428571429
          1.14285714285714
```

- Calcular el vector residuo de la solución y su norma 2. Comprobar que la solución de resolver las cuatro últimas ecuaciones tiene un mayor residuo.

```
res = 5×1
        -0.714285714285715
        -0.714285714285714
          1.42857142857143
         0.714285714285715
                         0

norma_res =
          1.88982236504614

norma_res_25 =
     5
```

- Ahora obtener la solución del sistema $A\mathbf{x} = \mathbf{b}$ utilizando los comandos de MATLAB: `\`, `linsolve()` y `lsqminnorm()`. ¿Qué observas?

```
x_backslash = 4×1
         0.142857142857143
          1.28571428571429
          1.28571428571429
          1.14285714285714

x_linsolve = 4×1
         0.142857142857143
          1.28571428571429
          1.28571428571429
          1.14285714285714

x_lsqminnorm = 4×1
         0.142857142857143
          1.28571428571429
          1.28571428571429
          1.14285714285714
```

Ejercicio 3. Ajuste con error cuadrático mínimo.

Determinar la función cuadrática que satisfaga al máximo (error cuadrático mínimo) la siguiente tabla:

x	0.8	10	12	16	20	40
y	0.88	1.22	1.64	2.72	3.96	11.96

- Escribir el sistema lineal asociado al problema, $A\mathbf{x} = \mathbf{b}$, especificar explícitamente A, $\mathbf{b}$ y calcular el rango de A.

```
A = 6×3
                    0.64                    0.8                      1
                     100                     10                      1
                     144                     12                      1
                     256                     16                      1
                     400                     20                      1
                    1600                     40                      1

b = 6×1
                    0.88
                    1.22
                    1.64
                    2.72
                    3.96
                   11.96

ans = 
     3
```

- Comprobar que el sistema de ecuaciones normales es compatible determinado.

```
rangoAtAextend = 
     3

compatible = logical
   1
```

- Calcular la solución $\mathbf{x}$ del sistema de ecuaciones normales y la norma 2 de su vector residuo.

```
x = 3×1
       0.00672424194360228
        0.0137689624389697
           0.7007863561967

ans = 3×1
       0.00672424194360229
         0.013768962438969
         0.700786356196706

n2_rx = 
         0.494263053142239
```

- Escribir la función cuadrática resultante con el comando `fprintf()` y el vector $\mathbf{x}$.

```
Función: 0.007x^2 +0.014x +0.701
```

Métodos iterativos

Ejercicio 4. Matrices de iteración

Determinar las matrices de iteración y vectores del método de Jacobi y del método de Gauss-Seidel y calcular la convergencia para el sistema $A\mathbf{x} = \mathbf{b}$ dado por:

$$A = \begin{pmatrix} 1 & 0.5 & 0.5 \\ 0.5 & 1 & 0.5 \\ 0.5 & 0.5 & 1 \end{pmatrix} \qquad \mathbf{b} = \begin{pmatrix} 4 \\ 1 \\ -4 \end{pmatrix}..$$

- Expresar la matriz A como la suma de tres matrices: $A = L + D + U$, tales que

$$\underbrace{\begin{pmatrix} 0 & 0 & \dots & 0 \\ a_{21} & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ a_{n1} & \dots & a_{nn-1} & 0 \end{pmatrix}}_{L} + \underbrace{\begin{pmatrix} a_{11} & 0 & \dots & 0 \\ 0 & a_{22} & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & a_{nn} \end{pmatrix}}_{D} + \underbrace{\begin{pmatrix} 0 & a_{12} & \dots & a_{1n} \\ 0 & 0 & \dots & \vdots \\ \vdots & \ddots & \ddots & a_{n-1n} \\ 0 & \dots & 0 & 0 \end{pmatrix}}_{U}.$$

(Nota: no hacen falta bucles, existen comandos de MATLAB como `diag(), tril() y triu()` que nos facilitan esta tarea.)

```
D = 3×3
     1     0     0
     0     1     0
     0     0     1

L = 3×3
                         0                         0                         0
                       0.5                         0                         0
                       0.5                       0.5                         0

U = 3×3
                         0                       0.5                       0.5
                         0                         0                       0.5
                         0                         0                         0
```

- Comprobar que el sistema es compatible determinado.

```
test = logical
   1
```

- Calcular la matriz B_J y el vector $\mathbf{c}_J$ del método de Jacobi : $\mathbf{x}^{(k+1)} = B_J\mathbf{x}^{(k)} + \mathbf{c}_J,\ k \geq 0$.

```
BJ = 3×3
                         0                      -0.5                      -0.5
                      -0.5                         0                      -0.5
                      -0.5                      -0.5                         0

cJ = 3×1
     4
     1
    -4
```

- Determinar la convergencia del método de Jacobi y la velocidad de convergencia.

```
rhoBJ =
                         1

convergencia = logical
   0

velocidad =
      2.22044604925031e-16
```

- Calcular la matriz B_J y el vector $\mathbf{c}_J$ del método de Gauss-Seidel: $\mathbf{x}^{(k+1)} = B_G\mathbf{x}^{(k)} + \mathbf{c}_G,\ k \geq 0$.

```
BG = 3×3
```

```
                        0                   -0.5                   -0.5
                        0                   0.25                  -0.25
                        0                  0.125                  0.375

cG = 3×1
                        4
                       -1
                     -5.5
```

- Determinar la convergencia del método de Gauss-Seidel y la velocidad de convergencia.

```
rhoBG = 
         0.353553390593274

convergencia = logical
   1

velocidad = 
          1.03972077083992
```

Ejercicio 5. Método de Gauss-Seidel

Escribir una función para el método iterativo de Gauss-Seidel.

function `[M,y] = GaussSeidel(A,b,x,eps,dif,n)`

M: Tabla con el número de iteración (columna 1) y el error como norma infinito del vector residuo (columna 2).

y: solución final tras convergencia.

A y b: matriz y vector del sistema.

x: valor inicial de la solución.

ε: error deseado de la solución.

dif : norma infinito de la diferencia entre pasos.

n: máximo número de iteraciones permitido.

Estructura propuesta:

- Comprobar que el sistema admite solución.
- Calcular las matrices de iteración: B_G , c_G.
- Calcular el radio espectral y comprobar que el método es convergente.
- Inicializar.
- Calcular en cada paso la norma infinito de la diferencia con la solución previa y la norma infinito del vector residuo.
- Iterar mientras el número de iteraciones $< n$, el error $\geq \varepsilon$ y la diferencia entre pasos sea $\geq dif$.
- Finalizar: guardar la solución final.

Aplicarla al sistema $A\mathbf{x} = \mathbf{b}$ dado por:

$$A = \begin{pmatrix} 2 & -1 & 0 & 1 \\ 1 & -2 & 1 & 1 \\ -1 & 1 & 4 & -1 \\ 1 & 0 & -1 & 2 \end{pmatrix}, \qquad \mathbf{b} = \begin{pmatrix} 1 \\ 1 \\ -1 \\ -2 \end{pmatrix}.$$

M1	M2
1	2
2	1.34375
3	0.3896484375
4	0.234100341796875
5	0.123820304870605
6	0.0394768416881561
7	0.0119591960683465
8	0.00313172934693284
9	0.0026575553101793
10	0.00113340426290165
11	0.000306337385253208
12	9.53682948949819e-05
13	4.61481980409406e-05
14	2.73296444763815e-05
15	9.50715289405046e-06
16	2.77842575235887e-06
17	6.65248303111099e-07
18	5.55226360665628e-07
19	2.58409952547112e-07
20	7.15368917436621e-08
21	2.30945738088906e-08
22	8.83382478278349e-09
23	5.93788551661589e-09
24	2.24982543706176e-09
25	6.35540953197733e-10
26	1.71576308716226e-10
27	1.13729470285762e-10
28	5.80866466037833e-11
29	1.77911019250132e-11
30	5.4873883215123e-12
31	1.61604063464438e-12
32	1.27076127398595e-12
33	5.24025267623074e-13

```
y = 4×1
     0.761904761905062
    -0.952380952381036
    -0.190476190476211
     -1.47619047619064
```

Documento preparado por Irene Parada, 13 de marzo de 2024