Mètodes Numèrics

Resolució de sistemes lineals

==========

=

TEMA 2 Àlgebra lineal numèrica

Table of Contents

EXERCICIS - Resolució de sistemas d'equacions lineals

1.- Nombre de condició d'una matriu

Càlcul del nombre de condició d'una matriu en Matlab®: consulteu condition number of matrix.

Estudieu com varia la solució del sistema lineal quan modifiquem la matriu del sistema

$(A + \Delta)(x + \delta x) = b$

`EXERCICI: 1) Consulteu la documentació de l'assignatura i feu una predicció de com petits canvis en A afecten a la solució x del sistema d'equacions Ax = b.`

Doncs bé, com és sabut, la solució d'un sistema d'equacions lineals no homogèni pot contenir errors que provinguin de diverses fonts. En un sistema d'equacions lineals, petits canvis en els coeficients(de la matriu A, del vector de termes independents o ambdós) poden provocar grans canvis en la solució, fent-la inestable. En el nostre cas, doncs, el problema raurà en el fet de tenir una matriu de coeficients mal condicionada. Per tant, el nostre sistema d'equacions lineals, construit amb la matriu A, haurà de ser definit com a sistema sensible, o bé també mal condicionat. És per això doncs, que podem predir que la nostra matriu A, tindrà un nombre de condició K(A) >= 1, ja que aquest és una mesura de l'estabilitat o condició de la matriu.

`EXERCICI: 2) Poseu a prova la vostra predicció pel vector b i les matrius A següents:`

$$(A|b) = \left(\begin{array}{cc|c} 1 & 1 & 100 \\ 1 & 0 & 1 \end{array}\right) \overset{sol.exacte}{\longrightarrow} \begin{pmatrix} 1 \\ 99 \end{pmatrix} \qquad (A|b) = \left(\begin{array}{cc|c} 1 & 1 & 100 \\ 1 & 0.01 & 1 \end{array}\right) \overset{sol.exacte}{\longrightarrow} \begin{pmatrix} ? \\ ? \end{pmatrix}$$

```
% Càlculs per fer: resoldre els dos sistemes i comparar les dues solucions obtingudes
A = [1 1; 1 0];
b = [100; 1];
sol = linsolve(A,b); %també sol=A\b
As = [1 1;1 0.01];
solExacteP =linsolve(As,b);

% En efecte, si comprobem amb Matlab el valor del nombre de condició de la matriu A, a
% és més major que 1, provocant que el nostre sistema sigui definit com a sensible, to
% i només havent variat un 0.01 el valor del coeficient (a22) de la matriu
% A. Cal notar també que en aquest cas, no hem perturbat cap valor del
% vector del terme independent.
cond(A); %K(A) > 1
%També podem calcular-ho fent:
res = norm(A)*norm(inv(A));

%Notem, els paràmetres de condicionament
k = cond(A,1);
rk = rcond(A); %en norma-1
table(k, rk, 'VariableNames',{'Nombre de condició','Nombre de condició recíproc'})
```

ans = 1×2 table

	Nombre de condició	Nombre de condició recíproc
1	4	0.3000

```
%Com a criteri de comparació entre la solució exacta x i la solució
%calculada x* = x + dx, del sistema lineal Ax=b, definirem el vector r(x*)
%per: r(x*)= Adx = Ax - Ax* = b - Ax*
rx = b - A*solExacteP; %vector residu ens mostra un mal condicionament del sistema
table(sol, solExacteP, 'VariableNames',{'Solució Exacte','Solució Matriu Pertorbada'})
```

ans = 2×2 table

	Solució Exacte	Solució Matriu Pertorbada
1	1	0
2	99	100

2.- Mètode dels mínims quadrats

Exemple. Les dades de la Taula 5.2 s'han tret de J.C. Miller & J.N. Miller (1993), *Statistics in Analytical Chemistry*, Ellis Horwood. Corresponen a una investigació sobre un test colorimètric per a la concentració de glucosa, en la que es varen obtenir absorbàncies per a sis concentracions patró de glucosa.

En els experiments de calibratge de l'anàlisi instrumental es pren sempre com a variable de control x la concentració (de fet, al ser una concentració patró, el seu valor no és experimental, sinó prefixat per l'usuari). La variable resposta y és en aquest cas, l'absorbància. Ajustem per mínims quadrats un model $y = a + bx$.

TAULA 5.2

Concentració (mM)	0	2	4	6	8	10
Absorbància	0.002	0.150	0.294	0.434	0.570	0.704

- Gràfica de les dades, correctament retolada

```
clearvars
x = [0:2:10]';    % Concentració(mM)
y = [0.002; 0.150; 0.294; 0.434; 0.570; 0.704]; % Absorbència
```

- Recta. La representació gràfica de les dades suggereix que la dependència de l'absorbància respecte a la concentració és lineal. Així, doncs, volem ajustar les dades a una recta y = a + bx.

EXERCICI: 1) obteniu els coeficients a i b de la recta.

```
p = polyfit(x,y,1);

%Els coeficients de la recta són:
a = p(1);
b = p(2);

table(a, b, 'VariableNames',{'Coeficient a','Coeficient b'})
```

ans = 1×2 table

	Coeficient a	Coeficient b
1	0.0701	0.0083

EXERCICI: 2) representeu les dades i la recta en un mateix gràfic. La gràfica ha d'estar perfectament retolada.

```
% Càlculs per fer
plot(x, b + a*x, 'DisplayName',sprintf('lineal'))
grid on
xlabel("Concentració en (mM)")
ylabel("Absorbància")
title("Test colorimètric per a la concentració de glucosa")
legend('show')

xlim([0.0 10.0])
```

```
ylim([0.00 0.80])
```

- Paràbola. Volem obtenir un model millor per mesurar l'absorbància en funció de la concentració de glucosa. Per exemple, proposem un model parabòlic y = a + bx + cx^2

EXERCICI: 3) obteniu els coeficients a, b i c de la paràbola.

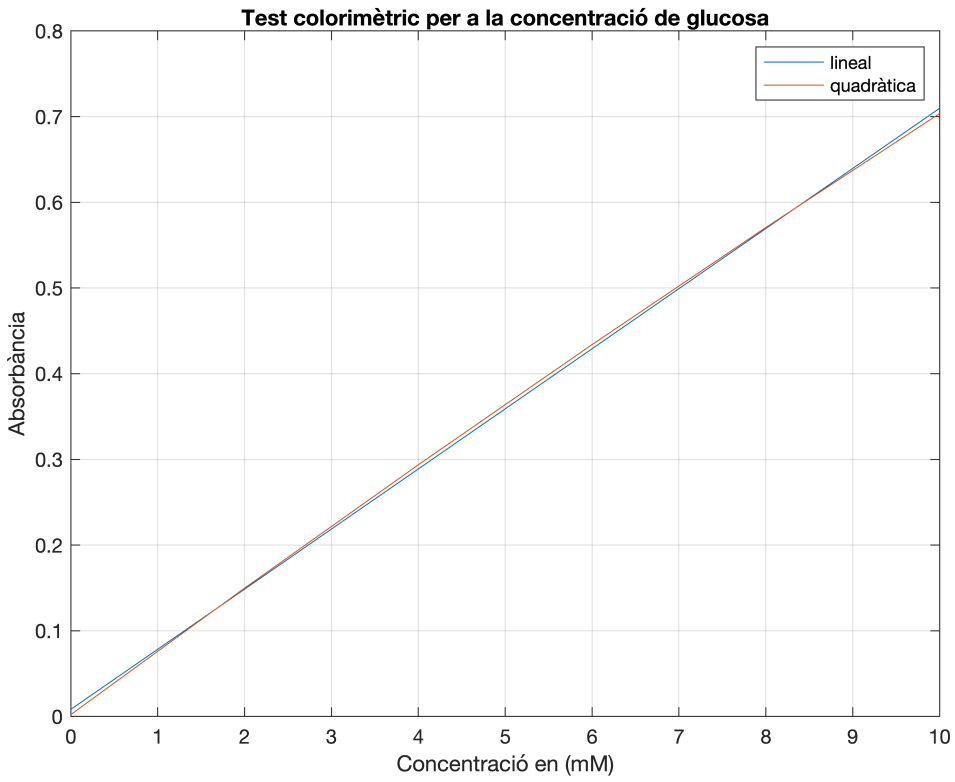
```
%Obtenim els coeficients de la paràbola
p1 = polyfit(x,y,2);

a = p1(1);
b = p1(2);
c = p1(3);
table(a, b, c, 'VariableNames',{'Coeficient a','Coeficient b','Coeficient c'})
```

ans = 1×3 table

	Coeficient a	Coeficient b	Coeficient c
1	-4.5536e-04	0.0747	0.0022

EXERCICI: 4) representeu les dades i la paràbola en un mateix gràfic. La gràfica ha d'estar perfectament retolada.

```
% %Càlculs per fer
% plottools
hold on
plot(x,c + b*x + a*x.^2, 'DisplayName',sprintf('quadràtica'))
hold off
```



3.- Mètodes iteratius

Exercici

Determineu el radi espectral de les matrius d'iteració dels mètodes de Jacobi i de Gauss-Seidel per resoldre el sistema lineal $Ax = b$ de valors:

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

Trobeu la solució del sistema lineal $Ax = b$ aplicant el mètode més ràpid (dels dos anteriors) fins que la diferència entre una iteració i la següent sigui inferior a $0.5 \cdot 10^{-8}$, comenceu amb $x^0 = (0,0,0,0,0)^\top$. Quantes iteracions són necessàries?

Doneu la longitud del vector residu cada 4 iteracions, doneu també el primer i darrer residu calculat.

```
% Càlculs per fer
A = [4 1 1 1 1; 1 4 1 1 1; 1 1 4 1 1; 1 1 1 4 1; 1 1 1 1 4];
b = [1; 1; 1; 1; 1];

D = diag(diag(A));
d = diag(1 ./ diag(A));

L = tril(A,-1);
U = triu(A,1);
```

```
EXERCICI: 1) Matriu d'iteració i radi del mètode de Jacobi
```

```
% Càlculs per fer
Bj = -d*(A-D); %matriu iteració
cj = d*b; %vector

rhoJ = max(abs(eig(Bj))); %radi espectral del mètode de Jacobi

if (rhoJ + eps >= 1) resp = "mètode de Jacobi divergent";
else resp = "mètode de Jacobi convergent";
end

jacobi = {rhoJ, resp}';
```

```
EXERCICI: 2) Matriu d'iteració i radi del mètode de Gauss-Seidel
```

```
% Càlculs per fer
dG = inv(L+D);
Bgs = -dG*U; %matriu iteració
```

```
cgs = dG*b; %vector

rhoGS = max(abs(eig(Bgs))); %radi espectral del mètode de Gauss-Seidel

if (rhoGS + eps >= 1) resp = "mètode de GAUSS-SEIDEL divergent";
else resp = "mètode de GAUSS-SEIDEL convergent";
end

gauss = {rhoGS, resp}';

table(jacobi, gauss, 'VariableNames',{'Jacobi','Gauss-Seidel'})
```

ans = 2×2 table

	Jacobi	Gauss-Seidel
1	1.0000	0.2509
2	"mètode de Jacobi divergent"	"mètode de GAUSS-SEIDEL convergent"

EXERCICI: 3) Iteracions i vector residu

```
%mètode de Jacobi
if rhoJ >= 1
    error('mètode de Jacobi divergent')
else %(rhoJ < 1)
    n = 10000; i = 0;
    x = zeros(size(b));
    tol = 0.5e-8;
    criteriParada(1) = 1;
    while(i < n && criteriParada(max(i,1)) > tol)
        i = i+1; y = Bj*x+cj;
        residu(i) = norm(b-A*y,'inf');
        e(i) = norm(y-x,'inf');
        delta(i) = norm(y-x,'inf');
        criteriParada(i) = delta(i)/norm(y,'inf');
        x = y;
    end
   table(residu',e',delta',criteriParada','VariableNames',{'residu','e^k','delta^k','s
   table(i, residu(1), residu(end), 'VariableNames',{'# iteracions', 'Primer Residu',
   x(1);
end
```

ans = 10000×4 table

	residu	e^k	delta^k	stop
1	1	0.2500	0.2500	1
2	1	0.2500	0.2500	Inf
3	1	0.2500	0.2500	1
4	1	0.2500	0.2500	Inf
5	1	0.2500	0.2500	1
6	1	0.2500	0.2500	Inf
7	1	0.2500	0.2500	1

	residu	e^k	delta^k	stop
8	1	0.2500	0.2500	Inf
9	1	0.2500	0.2500	1
10	1	0.2500	0.2500	Inf
11	1	0.2500	0.2500	1
12	1	0.2500	0.2500	Inf
13	1	0.2500	0.2500	1
14	1	0.2500	0.2500	Inf
15	1	0.2500	0.2500	1
16	1	0.2500	0.2500	Inf
17	1	0.2500	0.2500	1
18	1	0.2500	0.2500	Inf
19	1	0.2500	0.2500	1
20	1	0.2500	0.2500	Inf
21	1	0.2500	0.2500	1
22	1	0.2500	0.2500	Inf
23	1	0.2500	0.2500	1
24	1	0.2500	0.2500	Inf
25	1	0.2500	0.2500	1
26	1	0.2500	0.2500	Inf
27	1	0.2500	0.2500	1
28	1	0.2500	0.2500	Inf
29	1	0.2500	0.2500	1
30	1	0.2500	0.2500	Inf
31	1	0.2500	0.2500	1
32	1	0.2500	0.2500	Inf
33	1	0.2500	0.2500	1
34	1	0.2500	0.2500	Inf
35	1	0.2500	0.2500	1
36	1	0.2500	0.2500	Inf
37	1	0.2500	0.2500	1
38	1	0.2500	0.2500	Inf
39	1	0.2500	0.2500	1
40	1	0.2500	0.2500	Inf

	residu	e^k	delta^k	stop
41	1	0.2500	0.2500	1
42	1	0.2500	0.2500	Inf
43	1	0.2500	0.2500	1
44	1	0.2500	0.2500	Inf
45	1	0.2500	0.2500	1
46	1	0.2500	0.2500	Inf
47	1	0.2500	0.2500	1
48	1	0.2500	0.2500	Inf
49	1	0.2500	0.2500	1
50	1	0.2500	0.2500	Inf
51	1	0.2500	0.2500	1
52	1	0.2500	0.2500	Inf
53	1	0.2500	0.2500	1
54	1	0.2500	0.2500	Inf
55	1	0.2500	0.2500	1
56	1	0.2500	0.2500	Inf
57	1	0.2500	0.2500	1
58	1	0.2500	0.2500	Inf
59	1	0.2500	0.2500	1
60	1	0.2500	0.2500	Inf
61	1	0.2500	0.2500	1
62	1	0.2500	0.2500	Inf
63	1	0.2500	0.2500	1
64	1	0.2500	0.2500	Inf
65	1	0.2500	0.2500	1
66	1	0.2500	0.2500	Inf
67	1	0.2500	0.2500	1
68	1	0.2500	0.2500	Inf
69	1	0.2500	0.2500	1
70	1	0.2500	0.2500	Inf
71	1	0.2500	0.2500	1
72	1	0.2500	0.2500	Inf
73	1	0.2500	0.2500	1

	residu	e^k	delta^k	stop
74	1	0.2500	0.2500	Inf
75	1	0.2500	0.2500	1
76	1	0.2500	0.2500	Inf
77	1	0.2500	0.2500	1
78	1	0.2500	0.2500	Inf
79	1	0.2500	0.2500	1
80	1	0.2500	0.2500	Inf
81	1	0.2500	0.2500	1
82	1	0.2500	0.2500	Inf
83	1	0.2500	0.2500	1
84	1	0.2500	0.2500	Inf
85	1	0.2500	0.2500	1
86	1	0.2500	0.2500	Inf
87	1	0.2500	0.2500	1
88	1	0.2500	0.2500	Inf
89	1	0.2500	0.2500	1
90	1	0.2500	0.2500	Inf
91	1	0.2500	0.2500	1
92	1	0.2500	0.2500	Inf
93	1	0.2500	0.2500	1
94	1	0.2500	0.2500	Inf
95	1	0.2500	0.2500	1
96	1	0.2500	0.2500	Inf
97	1	0.2500	0.2500	1
98	1	0.2500	0.2500	Inf
99	1	0.2500	0.2500	1
100	1	0.2500	0.2500	Inf

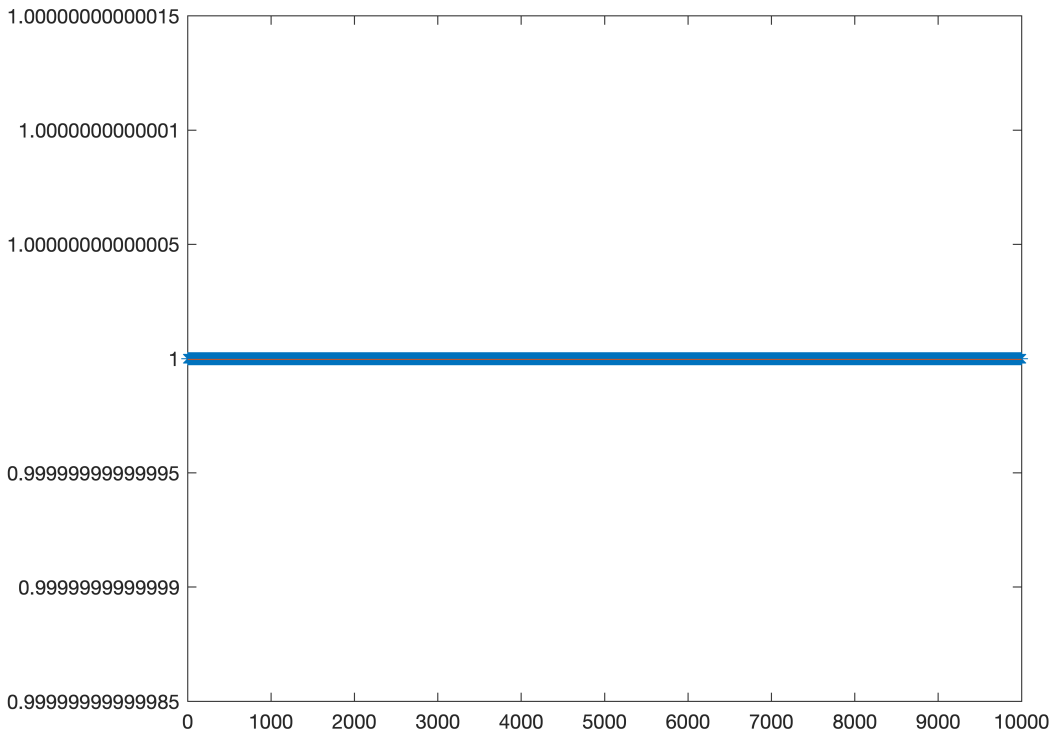
⋮

ans = 1×3 table

	# iteracions	Primer Residu	Darrer Residu
1	10000	1	1

```
%És interessant estudiar com evoluciona: delta^k/delta^k-1
ratio = delta(2:end) ./ delta(1:end-1);
plot(ratio,'-*')
hold on
```

```
plot(rhoJ * ones(size(ratio)))
hold off
```




```
%mètode de Gauss
clear residu criteriParada e delta
if rhoGS >=1
    error('mètode de gauss-seidel divergent')
else
    n = 10000; i = 0;
    x = zeros(size(b));
    tol = 0.5e-8;
    criteriParada(1) = 1;
    while(i<n && criteriParada(max(i,1)) > tol)
        i = i+1; y = Bgs*x+cgs;
        residu(i) = norm(b-A*y,'inf');
        e(i) = norm(y-x,'inf');
        delta(i) = norm(y-x,'inf');
        criteriParada(i) = delta(i)/norm(y,'inf');
        x = y;
    end
   table(residu',e',delta',criteriParada','VariableNames',{'residu','e^k','delta^k','s
   table(i, residu(1), residu(end), 'VariableNames',{'# iteracions', 'Primer Residu',
   x(1);
end
```

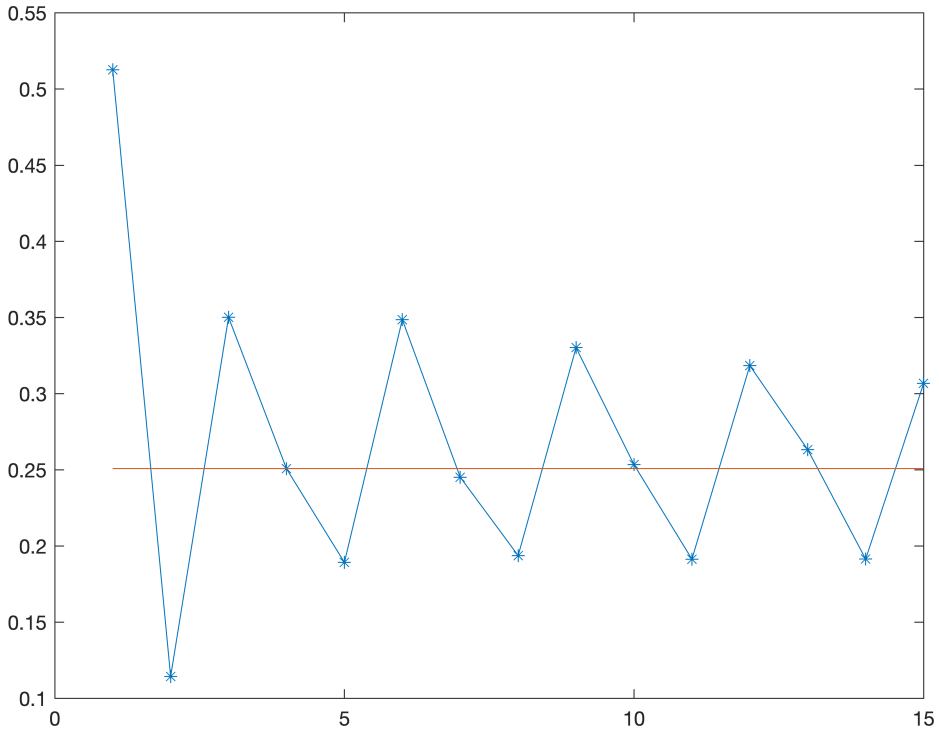
```
ans = 16×4 table
```

	residu	e^k	delta^k	stop
1	0.5127	0.2500	0.2500	1
2	0.0636	0.1282	0.1282	0.9232
3	0.0205	0.0147	0.0147	0.1144
4	0.0053	0.0051	0.0051	0.0409
5	0.0010	0.0013	0.0013	0.0103
6	0.0003	0.0002	0.0002	0.0019
7	0.0001	0.0001	0.0001	0.0007
8	0	0	0	0.0002
9	0	0	0	0
10	0	0	0	0
11	0	0	0	0
12	0	0	0	0
13	0	0	0	0
14	0	0	0	0
15	0	0	0	0
16	0	0	0	0

ans = 1×3 table

	# iteracions	Primer Residu	Darrer Residu
1	16	0.5127	3.4893e-10

```
%És interessant estudiar com evoluciona: delta^k/delta^k-1
ratio = delta(2:end) ./ delta(1:end-1);
plot(ratio,'-*')
hold on
plot(rhoGS * ones(size(ratio)))
hold off
```



Document preparat per M. Àngela Grau Gotés, 10 de març de 2022