UNIVERSIDAD NACIONAL DE SAN ANTONIO ABAD DEL CUSCO

FACULTAD DE CIENCIAS QUÍMICAS FÍSICAS Y MATEMÁTICAS

DEPARTAMENTO ACADÉMICO DE MATEMÁTICAS Y ESTADÍSTICA

UNSAAC

Lic. Guillermo Mario, Chuquipoma Pacheco

mariochuqui@hotmail.com

www.mariochuqui.webs.con

Métodos iterativos para sistemas lineales

Normas de vectores y matrices

Una norma vectorial en R^n es una función $\| \cdot \|$, de R^n en R con las siguientes propiedades:

(i) $\| \mathbf{x} \| \geq 0$ para todo $\mathbf{x} \in R^n$.

(ii) $\| \mathbf{x} \| = 0$ si y solo si $\mathbf{x} = (0,0,...,0)^t \equiv \mathbf{0}$.

(iii) $\| \alpha\mathbf{x} \| = |\alpha| \, \| \mathbf{x} \|$ para todo $\alpha \in R$ y $\mathbf{x} \in R^n$.

(iv) $\| \mathbf{x} + \mathbf{y} \| \leq \| \mathbf{x} \| + \| \mathbf{y} \|$ para todo $\mathbf{x}, \mathbf{y} \in R^n$.

Vector en R^n

El vector

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

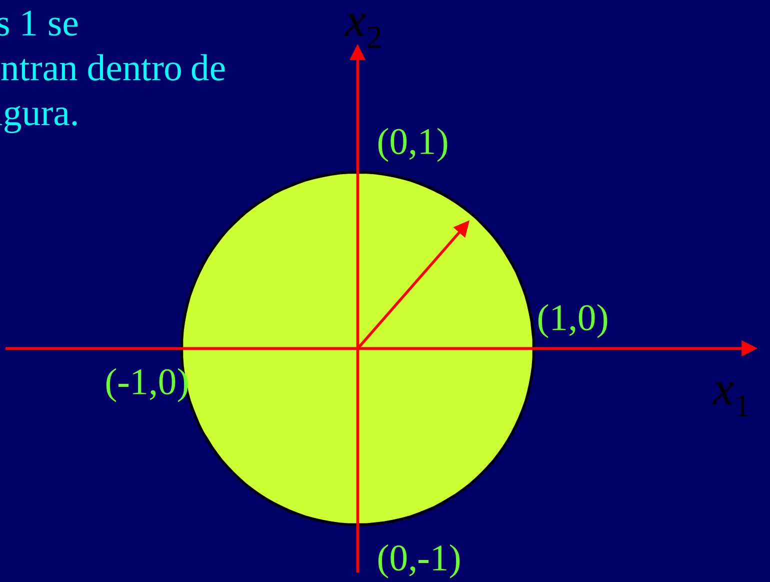
Se denotará por: $\mathbf{x} = (x_1, x_2, \ldots, x_n)^t$

Norma l_2 y l_∞

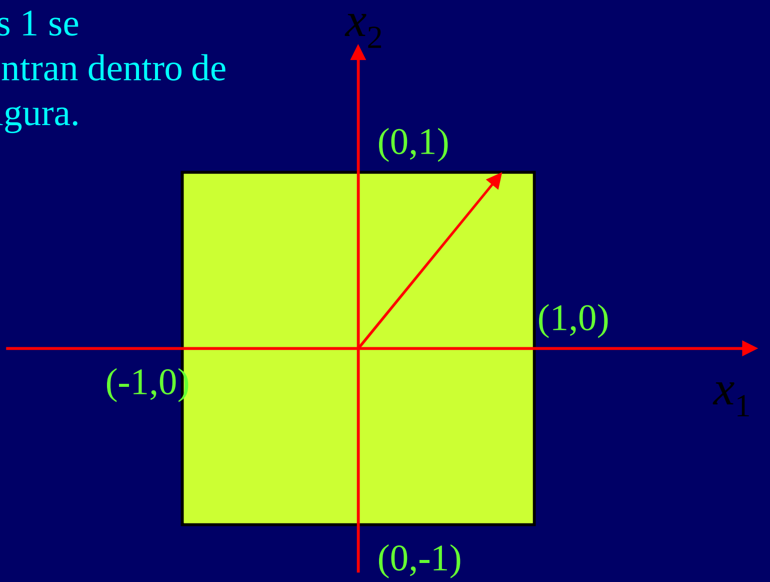
La norma l_2 o norma euclidiana se define como $\|\mathbf{x}\|_2 = \left\{ \sum_{i=1}^{n} x_i^2 \right\}^{1/2}$

La norma l_∞ se define como $\|\mathbf{x}\|_\infty = \max_{1 \le i \le n} |x_i|$

Los vectores en R^2 con la norma l_2 menos 1 se encuentran dentro de esta figura.



Los vectores en R^2 con la norme l_∞ menos 1 se encuentran dentro de esta figura.



Distancias

Si $\mathbf{x} = (x_1, x_2, \ldots, x_n)^t$ y $\mathbf{y} = (y_1, y_2, \ldots, y_n)^t$ son vectores en R^n las distancias l_2 y l_∞ éntre **x** y **y** están definidas por

$$\|\mathbf{x}-\mathbf{y}\|_2 = \left\{ \sum_{i=1}^{n} (x_i - y_i)^2 \right\}^{1/2}$$

$$\|\mathbf{x}-\mathbf{y}\|_\infty = \max_{1 \le i \le n} |x_i - y_i|$$

Ejemplo

$$3.3330x_1 + 15920x_2 - 10.333x_3 = 15913$$

$$2.2220x_1 + 16.710x_2 + 9.6120x_3 = 28.544$$

$$1.5611x_1 + 5.1791x_2 + 1.6852x_3 = 8.4254$$

Solución: $\mathbf{x} = (1.0000, 1.0000, 1.0000)^t$

Con Gauss: $\mathbf{x}' = (1.2001, 0.99991, 0.92538)^t$

$$\|\mathbf{x} - \mathbf{x}'\|_\infty = 0.2001$$

$$\|\mathbf{x} - \mathbf{x}'\|_2 = 0.21356$$

Convergencia

Se dice que una sucesión $\left\{\mathbf{x}^{(k)}\right\}_{k=1}^{\infty}$ de vectores en R^n converge a $\mathbf{x}$ respecto a la norma || . || si dado cualquier $\varepsilon > 0$, existe un entero $N(\varepsilon)$ tal que

$$\| \mathbf{x}^{(k)} - \mathbf{x}\| < \varepsilon \text{ para todo k} \geq N(\varepsilon)$$

La sucesión $\{\mathbf{x}^{(k)}\}$ converge a x respecto a $\| \, . \, \|_{\infty}$ si y solo si

$$\lim_{k\to\infty} x_i^{(k)} = x_i$$

Para cada $k = 1, 2, ..., n$.

Para todo $x \in R^n$,

$$\|x\|_{\infty} \leq \|x\|_2 \leq \sqrt{n}\|x\|_{\infty}$$

Norma matricial

Una norma matricial sobre el conjunto de todas las matrices n x n es una función $\| \cdot \|$, definida en ese conjunto y que satisface para todas las matrices A, B de n x n y todos los números α:

(i) $\| A \| \geq 0$

(ii) $\| A \| = 0$ si y solo si A es 0.

(iii) $\| \alpha A \| = |\alpha| \, \| A \|$

(iv) $\| A + B \| \leq \| A \| + \| B \|$

(v) $\| AB \| \leq \| A \| \, \| B \|$

Una distancia entre A y B es $\| A - B \|$

Norma matricial l_2 y l_∞

Norma l_∞

$$\left\|A\right\|_\infty = \max_{\|\mathbf{x}\|_\infty = 1} \| A\mathbf{x} \|$$

Norma l_2

$$\left\|A\right\|_2 = \max_{\|\mathbf{x}\|_2 = 1} \| A\mathbf{x} \|$$

Cálculo de la norma l_∞

Norma l_∞ se calcula con

$$\|A\|_\infty = \max_{1 \le i \le n} \left| \sum_{j=1}^{n} |a_{ij}| \right|$$

Cálculo de la norma l_2

Norma l_2 se calcula con

$$\|\mathbf{A}\|_2 = \sqrt{\rho\left(\mathbf{A}^t\mathbf{A}\right)}$$

Donde ρ es el *radio espectral* de $\mathbf{A}^t\mathbf{A}$.

El radio espectral es el máximo de los valores de propios de la matriz, calculadas con la ecuación

$$\det\left(\mathbf{A}^t\mathbf{A} - \lambda\mathbf{I}\right) = 0$$

Ejemplo

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 2 & 1 \\ -1 & 1 & 2 \end{bmatrix}$$

$$\mathbf{A}^t\mathbf{A} = \begin{bmatrix} 1 & 1 & -1 \\ 1 & 2 & 1 \\ 0 & 1 & 2 \end{bmatrix}\begin{bmatrix} 1 & 1 & 0 \\ 1 & 2 & 1 \\ -1 & 1 & 2 \end{bmatrix} = \begin{bmatrix} 3 & 2 & -1 \\ 2 & 6 & 4 \\ -1 & 4 & 5 \end{bmatrix}$$

$$\det\left(\mathbf{A}^t\mathbf{A} - \lambda\mathbf{I}\right) = \begin{vmatrix} 3-\lambda & 2 & -1 \\ 2 & 6-\lambda & 4 \\ -1 & 4 & 5-\lambda \end{vmatrix} = 0$$

o

$$-\lambda^3 + 14\lambda^2 - 42\lambda = 0$$

Resolviendo se encuentra

$$||\mathbf{A}||_2 = 3.16$$

Normas en Matlab

norm (*a*, *p*)

Computa la norm p de la matriz *a*. Si no hay segundo argumento, se supone p = 2.

Si *a* es una matriz:

p = 1 norma 1, la suma de la columna mayor de los valores absolutos de a.
p = 2 El mayor valor singular de *a*.
p = Inf norma Infinita, la suma del renglón de los valores absolutos de *a*.
p = "fro" la norm Frobenius de *a*, sqrt (sum (diag (*a*' * *a*))).

Métodos iterativos

Los métodos iterativos se utilizan para sistemas de un gran número de ecuaciones con un alto porcentaje de elementos cero.

Un método de solución de

$$A\mathbf{x} = \mathbf{b}$$

Comienza con una aproximación $\mathbf{x}^{(0)}$ y genera la sucesión de vectores

$$\{\mathbf{x}^{(k)}\} \qquad k = 0 \,..\, \infty$$

Método de solución

El sistema $A\mathbf{x} = \mathbf{b}$ se escribe de la forma

$$\mathbf{x} = T\mathbf{x} + \mathbf{c}$$

Donde T es una matriz fija y un vector $\mathbf{c}$.

La sucesión de vectores se genera calculando

$$\mathbf{x}^{(k)} = T\,\mathbf{x}^{(k-1)} + \mathbf{c}$$

Ejemplo

$$
\begin{aligned}
&E1:\ 10x_1 - x_2 + 2x_3 = 6\\
&E2:\ -x_1 + 11x_2 - x_3 + 3x_4 = 25\\
&E3:\ 2x_1 - x_2 + 10x_3 - x_4 = -11\\
&E4:\ 3x_2 - x_3 + 8x_4 = 15
\end{aligned}
$$

Despejando x_i de la ecuación i:

$$
\begin{aligned}
x_1 &= 1/10x_2 - 1/5x_3 + 3/5\\
x_2 &= 1/11x_1 + 1/11x_3 - 3/11x_4 + 25/11\\
x_3 &= -1/5x_1 + 1/10x_2 + 1/10x_4 - 11/10\\
x_4 &= -3/8x_2 + 1/8x_3 + 15/8
\end{aligned}
$$

$$T = \begin{bmatrix} 0 & \frac{1}{10} & -\frac{1}{5} & 0 \\ \frac{1}{11} & 0 & \frac{1}{11} & -\frac{3}{11} \\ -\frac{1}{5} & \frac{1}{10} & 0 & \frac{1}{10} \\ 0 & -\frac{3}{8} & \frac{1}{8} & 0 \end{bmatrix}$$

$$c = \begin{bmatrix} \frac{3}{5} \\ \frac{25}{11} \\ -\frac{11}{10} \\ \frac{15}{8} \end{bmatrix}$$

k	0	1	2	3	4	5	6	7	8	9	10
x1	0.0000	0.6000	1.0473	0.9326	1.0152	0.9890	1.0032	0.9981	1.0006	0.9997	1.0001
x2	0.0000	2.2727	1.7159	2.0533	1.9537	2.0114	1.9922	2.0023	1.9987	2.0004	1.9998
x3	0.0000	-1.1000	-0.8052	-1.0493	-0.9681	-1.0103	-0.9945	-1.0020	-0.9990	-1.0004	-0.9998
x4	0.0000	1.8750	0.8852	1.1309	0.9738	1.0214	0.9944	1.0036	0.9989	1.0006	0.9998

Tarea

Resuelva los siguientes sistemas usando el método iterativo de Jacobi.

$$\begin{aligned} 3x_1 - x_2 + x_3 &= 1 \\ 3x_1 + 6x_2 + 2x_3 &= 0 \\ 3x_1 + 3x_2 + 7x_3 &= 4 \end{aligned}$$

$$\begin{aligned} 10x_1 - x_2 \qquad &= 0 \\ -x_1 + 10x_2 - 2x_3 &= 7 \\ - 2x_2 + 10x_3 &= 6 \end{aligned}$$

$$\begin{aligned} 10x_1 + 5x_2 \qquad\qquad &= 6 \\ 5x_1 + 10x_2 - 4x_3 \qquad &= 25 \\ - 4x_2 + 8x_3 - x_4 &= -11 \\ - x_3 + 5x_4 &= -11 \end{aligned}$$

$$\begin{aligned} 4x_1 + x_2 - x_3 + x_4 &= -2 \\ x_1 + x_2 - x_3 x_4 &= -1 \\ -x_1 - x_2 + 5x_3 + x_4 &= 0 \\ x_1 - x_2 + x_3 + 3x_4 &= 1 \end{aligned}$$

Método iterativo de Jacobi

Se resuelve la i-ésima ecuación para x_i:

$$x_i = \sum_{\substack{j=1 \\ j \neq i}}^{n} \left(-\frac{a_{ij} x_j}{a_{ii}} \right) + \frac{b_i}{a_{ii}}$$

Siempre que $a_{ii} \neq 0$

$$x_i^{(k)} = \frac{\sum_{\substack{j=1 \\ j \neq i}}^{n} \left(-a_{ij} x_j^{(k-1)} \right) + b_i}{a_{ii}}$$

Método iterativo de Jacobi

Escribimos la matriz del sistema como

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix}$$

$$= \begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & a_{nm} \end{bmatrix} - \begin{bmatrix} 0 & 0 & \cdots & 0 \\ -a_{21} & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ -a_{n1} & -a_{n2} & \cdots & 0 \end{bmatrix} - \begin{bmatrix} 0 & -a_{12} & \cdots & -a_{1m} \\ 0 & 0 & \cdots & -a_{2m} \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}$$

$$A = D - L - U$$

Método iterativo de Jacobi

$$A\mathbf{x} = \mathbf{b}$$

$$(D - L - U)\mathbf{x} = \mathbf{b}$$

$$D\mathbf{x} = (L + U)\mathbf{x} + \mathbf{b}$$

$$\mathbf{x} = D^{-1}(L + U)\mathbf{x} + D^{-1}\mathbf{b}$$

$$\mathbf{x}^{(k)} = D^{-1}(L + U)\mathbf{x}^{(k-1)} + D^{-1}\mathbf{b} \qquad k = 1, 2, \ldots$$

$$\mathbf{x}^{(k)} = T_j\mathbf{x}^{(k-1)} + \mathbf{c}_j$$

Código en matlab

```
function [x,J,c] = jacobi(A,b,n,z)
%       n -- número de iteraciones
%       z -- vector inicial(default 0)
%       x -- solución final
%       J -- matriz de Jacobi
%       c -- vector de Jacobi
if nargin <=3, z=0*b; end
D = diag(diag(A));
J = D\(D - A);
c = D\b;
x=z;
for k = 1:n
   x = J*x + c;
   fprintf(1,'%3d      ',k)
   fprintf(1,'%5.4f      ',x')
   fprintf(1,'\n')
end
```

Método iterativo de Jacobi

Entrada: número de ecuaciones, elementos a_{ij}, los elementos b_i, X0 los valores iniciales de x, Tol la tolerancia, el número de iteraciones N.

Salida: la solución aproximada, o el mensaje de que no hay solución.

```
1. k = 1
2. Mientras k <= N hacer pasos 3-9
3.     Para i = 1 hasta N
4.
```

$$x_i^{(k)} = \frac{\sum_{\substack{j=1 \\ j \neq i}}^{n} \left(-a_{ij} x_j^{(k-1)} \right) + b_i}{a_{ii}}$$

```
5.     Si ||x - x0|| < TOL
6.        Regresar (x1, x2, ..., xn) y parar
7.     k = k +1
8.     Para j = 1 hasta N
9.        X0j = xj
10. Salida("no se encontro solución")
```

Método de Jacobi en Matlab

```
function y = jacobi(a,b,x0,tol,maxi)
[n muda] = size(a);
k = 1;
x = x0;
while k<=maxi
    for i=1:n
        suma = 0;
        for j=1:n
            if i~=j
                suma = suma - a(i,j)*x(j);
            end
        end
        x(i) = (suma+b(i))/a(i,i);
    end
    if norm(x-x0)<tol
        y = x;
        return
    end
    k = k + 1;
    x0 = x;
end
fprintf('No se  encontró solución')
```

Ejemplo

E1: $10x_1 - x_2 + 2x_3 = 6$

E2: $-x_1 + 11x_2 - x_3 + 3x_4 = 25$

E3: $2x_1 - x_2 + 10x_3 - x_4 = -11$

E4: $3x_2 - x_3 + 8x_4 = 15$

```
>> a=[10,-1,2,0;-1,11,-1,3;2,-1,10,-1;0,3,-1,8];
>> b = [6,25,-11,15]';
>> jacobi(a,b,[0,0,0,0],1e-3,20)

ans =

    1.0001    2.0000   -1.0000    1.0000

>>
```

Método iterativo de Gauss-Seidel

En el paso k se utilizan las x_i ya calculadas.

$$x_i^{(k)} = \frac{-\sum_{j=1}^{i-1}\left(a_{ij}x_j^{(k)}\right) - \sum_{j=i+1}^{n}\left(a_{ij}x_j^{(k-1)}\right) + b_i}{a_{ii}}$$

Excepto por esta fórmula, el algoritmo es el mismo que el de Jacobi.

Gauss-Seidel en Matlab

```
function [x,G,c] = gsmp(A,b,n,z)
%       n -- número de iteraciones
%       z -- vector inicial (default 0)
%       x -- iteración final
%       G -- matriz Gauss-Seidel
%       c -- vector Gauss-Seidel
if nargin <=3, z=0*b; end
LD = tril(A);
G = -LD\triu(A,1);
c = LD\b;
x=z;
for i = 1:n
   x = G*x + c;
   fprintf(1,'%3d     ',i)
   fprintf(1,'%5.5f     ',x')
   fprintf(1,'\n')
end
```

Ejemplo

```
a =
   4   3   0
   3   4  -1
   0  -1   4
b =
   24
   30
  -24
gsmp(a,b,7,[1 1 1]')
  1     5.25000     3.81250     -5.04688
  2     3.14062     3.88281     -5.02930
  3     3.08789     3.92676     -5.01831
  4     3.05493     3.95422     -5.01144
  5     3.03433     3.97139     -5.00715
  6     3.02146     3.98212     -5.00447
  7     3.01341     3.98882     -5.00279
```

Métodos de relajación

Los métodos de relajación tiene el siguiente esquema.

$$x_i^{(k)} = \frac{w\left(-\sum_{j=1}^{i-1}\left(a_{ij}x_j^{(k)}\right) - \sum_{j=i+1}^{n}\left(a_{ij}x_j^{(k-1)}\right) + b_i\right)}{a_{ii}} + (1-w)\,x_i^{(k-1)}$$

Si $0 < w < 1$ se llama subrelajación y se utiliza cuando el método de Gauss-Seidel no converge.

Si $1 < w$ se llama sobrerrelajación y sirve para acelerar la convergencia. Valores típicos de 1.2 a 1.7.

Métodos de relajación

$$a_{ii}x_i^{(k)} + w\sum_{j=1}^{i-1}\left(a_{ij}x_j^{(k)}\right) = a_{ii}\left(1-w\right)x_i^{(k-1)} - w\sum_{j=i+1}^{n}\left(a_{ij}x_j^{(k-1)}\right) + wb_i$$

En forma matricial

$$(D + wL)\mathbf{x} = [(1 - w)D - wU]\mathbf{x} + w\mathbf{b}$$

$$\mathbf{x} = (D + wL)^{-1}[(1 - w)D - wU]\mathbf{x} + (D + wL)^{-1}w\mathbf{b}$$

Relajación en Matlab

```
function [x,G,c] = relaj(A,b,n,w,z)
%       n -- número de iteraciones
%       z -- vector inicial (default 0)
%       x -- iteración final
%       G -- matriz Gauss-Seidel
%       c -- vector Gauss-Seidel
%       w -- Coeficiente de relajación
if nargin <=3, z=0*b; end
D = diag(diag(A));
LD = inv(D+w*tril(A,-1));
G = LD*((1-w)*D-w*triu(A,1));
c = w*LD*b;
x=z;
for i = 1:n
   x = (G*x + c);
   fprintf(1,'%3d     ',i)
   fprintf(1,'%5.5f     ',x')
   fprintf(1,'\n')
end
```

Ejemplo

```
a =
   4   3   0
   3   4  -1
   0  -1   4
b =
   24
   30
  -24

relaj(a,b,7,1.25,[1 1 1]')
  1      6.31250     3.51953     -6.65015
  2      2.62231     3.95853     -4.60042
  3      3.13330     4.01026     -5.09669
  4      2.95705     4.00748     -4.97349
  5      3.00372     4.00292     -5.00571
  6      2.99633     4.00093     -4.99828
  7      3.00005     4.00026     -5.00035
```

Tarea

Resuelva los problemas anteriores por Gauss Seidel y relajación


gracias

Lic. Guillermo Mario Chuquipoma Pacheco
mariochuqui@hotmail.com
http://www.mariochuqui.webs.com

UNSAAC