

ELEMENTOS DE CÁLCULO NUMÉRICO (M) - CÁLCULO NUMÉRICO
Segundo Cuatrimestre 2014

Práctica de Programación - Octave - Matlab.

Ejercicio 1 Matrices aleatorias: Leer el resultado de `help rand`, `help randn` y de `help hist`. Luego observar el resultado de los siguientes comandos:

- | | | |
|--------------------------------|---------------------------------|----------------------------------|
| a) <code>rand(4)</code> | d) <code>y=rand(1000);</code> | g) <code>hist(x)</code> |
| b) <code>x=rand(1,100);</code> | e) <code>hist(y)</code> | h) <code>y=randn(1,10000)</code> |
| c) <code>hist(x)</code> | f) <code>x=randn(1,1000)</code> | i) <code>hist(y)</code> |

Ejercicio 2 Generar los siguientes vectores:

- a) Una matriz de 3×5 de números aleatorios uniformemente distribuidos entre 0 y 1.
- b) Un vector columna de 100 números aleatorios enteros uniformemente distribuidos entre 0 y 99. (Consultar `help floor`).
- c) Un vector fila de 20 números aleatorios uniformemente distribuidos entre -1 y 1.

Ejercicio 3 Normas y condicionamiento: Ingresar los siguientes comandos e interpretar el resultado.

- | | | |
|---------------------------------------|---------------------------------------|----------------------------------|
| a) <code>x=[1 2 -1 0 2];</code> | f) <code>inv(A)</code> | k) <code>norm(B,Inf)</code> |
| b) <code>norm(x)</code> | g) <code>norm(A)</code> | l) <code>norm(inv(B),1)</code> |
| c) <code>norm(x,1)</code> | h) <code>cond(A)</code> | m) <code>norm(inv(B),Inf)</code> |
| d) <code>norm(x,Inf)</code> | i) <code>B=[1 2 3;1 3 5;2 1 4]</code> | n) <code>cond(B,1)</code> |
| e) <code>A=[2 0 0;0 3 0;0 0 5]</code> | j) <code>norm(B,1)</code> | o) <code>cond(B,Inf)</code> |

Ejercicio 4 Resolución de sistemas: Ingresar los siguientes comandos e interpretar el resultado.

- | | |
|---------------------------------------|-------------------------|
| • <code>A=[1 4 5;1 3 1;2 0 4];</code> | • <code>inv(A)*b</code> |
| • <code>b=[1;-1;3];</code> | • <code>A\b</code> |
| • <code>det(A)</code> | • <code>b'/A'</code> |

Ejercicio 5 Matrices tridiagonales: Ingresar los siguientes comandos e interpretar el resultado.

- | | | |
|--------------------------------|---|----------------------------|
| a) <code>x=2*ones(10,1)</code> | e) <code>A = A + diag(y,1)</code> | i) <code>diag(B)</code> |
| b) <code>A=diag(x)</code> | f) <code>diag(y,-1)</code> | j) <code>diag(B,1)</code> |
| c) <code>y=ones(9,1)</code> | g) <code>A = A + diag(y,-1)</code> | k) <code>diag(B,-2)</code> |
| d) <code>diag(y,1)</code> | h) <code>B = [1 1 1;2 3 1;0 2 9]</code> | |

Ejercicio 6 Tiempo de ejecución: Consultar `help tic` y `help toc`.

Para resolver el sistema $Ax = b$ se dispone de dos procedimientos: `x=inv(A)*b` y `x=A\b`. Se desea determinar cuál de los dos es más eficiente. Para ello se propone el siguiente experimento:

- Generar una matriz aleatoria A de 5000×5000 y un vector aleatorio b de 5000×1 .
- Introducir la siguiente secuencia de comandos: `tic; inv(A)*b; toc`. Puede escribirse todo en una misma línea antes de presionar Enter, para evitar consumir tiempo de reloj escribiendo.
- Introducir la siguiente secuencia de comandos: `tic; A\b; toc`.
- Repetir el experimento un par de veces. ¿Qué método de resolución es mejor?

(Obs.: En el primer método, la operación costosa es el cálculo de `inv(A)`, no el producto `inv(A)*b`. Para verificarlo puede realizarse el experimento en dos etapas: `B=inv(A)` y luego `B*b`, calculando el tiempo de cada etapa.).