ENTREGA 1 y 2

Apellidos, Nombre:	Abengózar Vilar, Diego José
Apellidos, Nombre:	García Castellanos, Alejandro

CASO 1 (Caso de prueba): MATRIZ CONSISTENTE. w → M → w. Análisis. Resultados

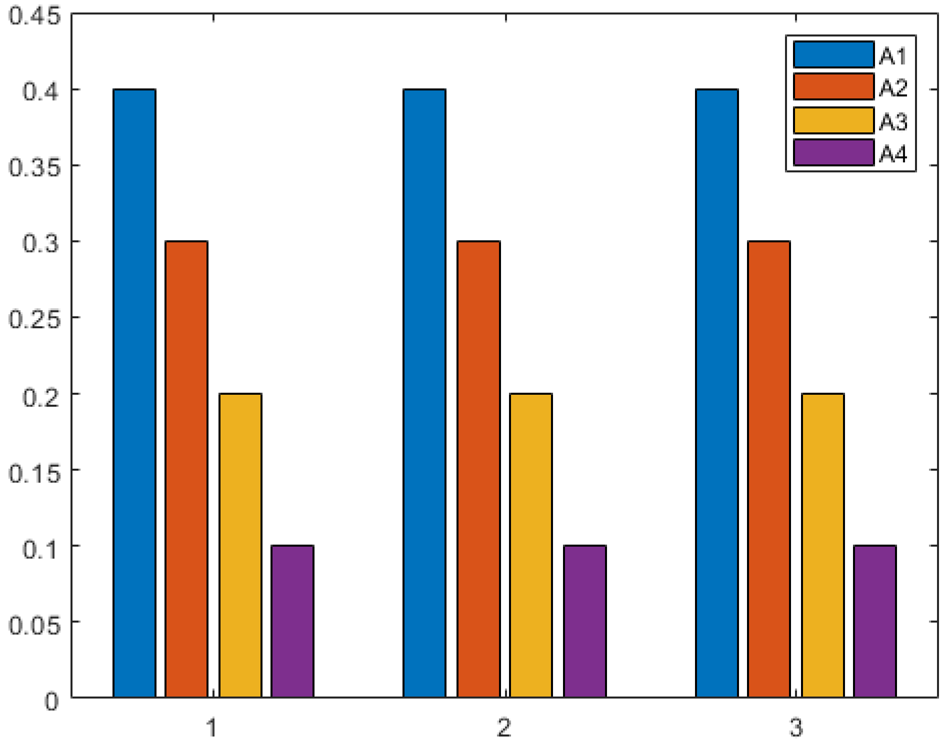
Código:

```
clc;
clear;
fprintf("-------------Consistente-------------\n");
w = [0.4 0.3 0.2 0.1]
M= zeros(4);
for i = 1:4
    for j = 1:4
        M(i,j) = w(i)/w(j);
    end
end

M
ic(M)
fprintf("-------------Metodo de la potencia-------------\n");
w0 = funciones(M, 0) % potencia(M)
[errorInf, IndexMaxErr, errorFro, errorUno, errorErrRel] = errores(M,w0)

fprintf("-------------Metodo de Minimos Cuadrados [Log]-------------\n");
w1 = funciones(M, 1) % minCuadLog(M)
[errorInf, IndexMaxErr, errorFro, errorUno, errorErrRel] = errores(M,w1)
fprintf("-------------Metodo de Minimos Cuadrados [Ponderado]-------------\n");
w2 = funciones(M, 2) % minCuadPond(M)
[errorInf, IndexMaxErr, errorFro, errorUno, errorErrRel] = errores(M,w2)

%Dibujar pesos
figure();
bar([w0';w1';w2']);
l=compose("A%d",(1:length(M)));
legend(l);
```

Tabla de resultados y medidas de error

IC(M)= 0	w	Ranking	Max Residuo	Error Fr	Error Rel.	Error Norm1
MÉTODO 1	(0.4000,0.3000,0.2000,0.1000)	(A1, A2, A3, A4)	8.8818e-16	1.0489e-15	5.8835e-16	8.8818e-16
MÉTODO 2	(0.4000,0.3000,0.2000,0.1000)	(A1, A2, A3, A4)	8.8818e-16	1.0551e-15	4.7175e-16	1.3323e-15
MÉTODO 3	(0.4000,0.3000,0.2000,0.1000)	(A1, A2, A3, A4)	1.3323e-15	2.6210e-15	1.7297e-15	3.3307e-15

Como la matriz es consistente podemos observar que al ser el caso ideal los tres métodos nos dan el mismo resultado y contienen errores bastantes parecidos y prácticamente despreciables, ya que se aproximan a la precisión máxima de la máquina. También se puede observar que estos errores son tan bajos que dependiendo de la ejecución pueden variar, dando situaciones donde se considere que el error es 0,0.



CASO 2. Un caso no consistente. Por ejemplo

Código:

```
clc;
clear;
fprintf("-------------NO Consistente-------------\n");
M=[1 2,2,4;1,1,1.5,3;0.5,0.6,1,2;0.25,0.3,0.5,1]
ic(M)
fprintf("-------------Metodo de la potencia-------------\n");
w0 = funciones(M, 0) % potencia(M)
[errorInf, IndexMaxErr, errorFro, errorUno, errorErrRel] = errores(M,w0)

fprintf("-------------Metodo de Minimos Cuadrados [Log]-------------\n");
w1 = funciones(M, 1) % minCuadLog(M)
[errorInf, IndexMaxErr, errorFro, errorUno, errorErrRel] = errores(M,w1)
fprintf("-------------Metodo de Minimos Cuadrados [Ponderado]-------------\n");
w2 = funciones(M, 2) % minCuadPond(M)
[errorInf, IndexMaxErr, errorFro, errorUno, errorErrRel] = errores(M,w2)

%Dibujar pesos
figure();
bar([w0';w1';w2']);
l=compose("A%d",(1:length(M)));
legend(l);
```

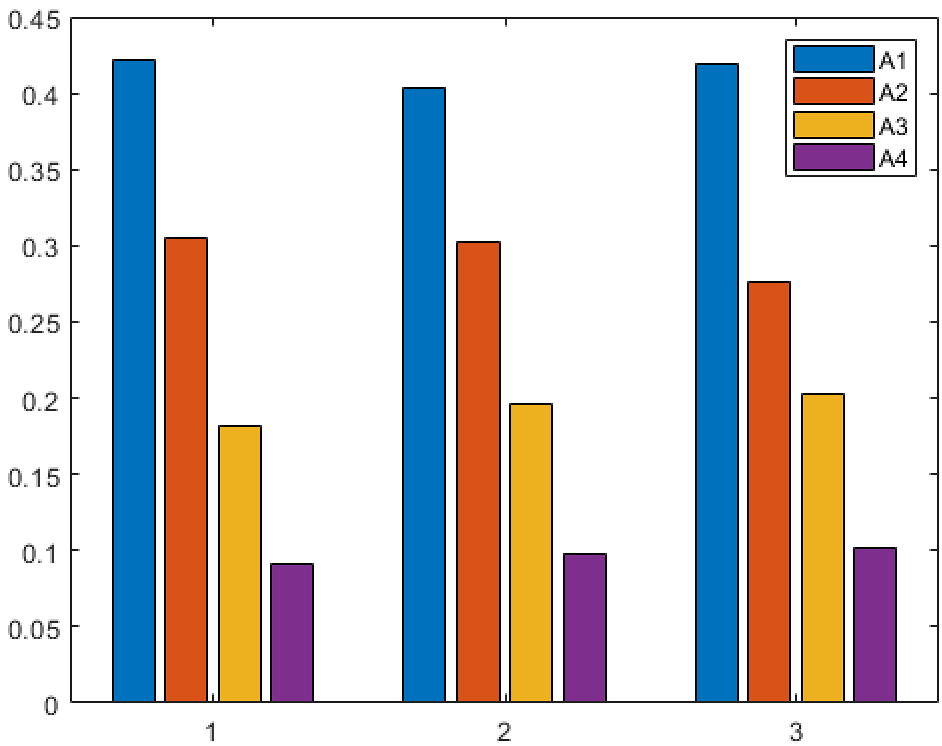
Tabla de resultados y medidas de error

IC(M)= 0.0563	w	Ranking	Max Residuo	Error Fr	Error Rel.	Error Norm1
MÉTODO 1	(0.4221,0.3053,0.1818,0.0909)	(A1, A2, A3, A4)	0.6446	1.0697	0.5395	1.0037
MÉTODO 2	(0.4037,0.3018,0.1963,0.0982)	(A1, A2, A3, A4)	0.6625	0.7274	0.4380	0.7380
MÉTODO 3	(0.4195,0.2766,0.2026,0.1014)	(A1, A2, A3, A4)	0.4832	0.6986	0.5418	0.6824

Como podemos observar al no tener un índice de consistencia igual a cero sabemos que la matriz no es consistente, y esto da lugar a los resultados obtenidos. Vemos que en este caso el vector de pesos varia pero la perturbación respecto a la matriz consistente no es lo suficientemente grande para que el ranking (independientemente del método usado) cambie, al ser su IC bastante cercano al cero.

También es necesario destacar que el método de la potencia es el que, en general, tiene mayor error; esto puede ser debido a que el método de mínimo cuadrados intentan minimizar el error entre la matriz M y la W, donde $w_{i,j}=w_i/w_j$, mientras que el método de la potencia intenta minimizar el error del cálculo del autovalor dominante y su correspondiente autovector.

Además, al linealizar el sistema para poder aplicar el método de mínimos cuadrados nos damos con la situación de que no estamos minimizando realmente el problema original, sino el original una vez aplicado la transformación correspondiente. Es por esto por lo que los dos métodos de mínimos cuadrados difieren las soluciones uno de otro; y sobre todo podemos observar que es mayor el del logaritmo.



ANEXO:

funciones.m

```
function w = funciones(M, metodo)

if nargin == 1
    [~, w, ~] = potencia(M);
elseif nargin == 2
    if metodo == 0
        [~, w, ~] = potencia(M);
    elseif metodo == 1
        w = minCuadLog(M);
    else
        w = minCuadPond(M);
    end
else
    fprintf("Número erróneo de argumentos\n");
end

return
```

potencia.m

```
function [lambda,x,iter] = potencia(A,tol,nmax,x0)
% Calcula el mayor (abs) autovalor lambda de A y un autovector asociado x

n=size(A,1);

if nargin == 1
    tol = 1e-06; % Tolerancia
    x0=rand(n,1); % Vector de arranque
    nmax=100; % NÂº mÃ¡x iteraciones
end

x0=x0/norm(x0); x1=A*x0;
lambda=x0'*x1;
err = tol*abs(lambda) + 1;
iter=0;

while err > tol*abs(lambda) && abs(lambda) ~= 0 && iter <= nmax
    x=x1; x=x/norm(x);
    x1=A*x; lambda_new=x'*x1;
    err = abs(lambda_new - lambda);
    lambda=lambda_new;
    iter = iter + 1;
end

x = x / sum(x);

end
```

minCuadLog.m

```
function w = minCuadLog(M)

s=size(M);
long = (s(1)*s(2))-s(1);

H = zeros([long,s(2)]);
b=zeros([long,1]);

k=1;
for i= (1:s(1))
    for j=1:s(2)
        if (i ~= j)
            % Generar H
            H(k,i)=1;
            H(k,j)=-1;
            % Generar b
            b(k)=log(M(i,j));
            k=k+1;
        end
    end
end

% Resolver con minimos cuadrados
v=H\b;
% Deshacer el cambio de logaritmo.
w=exp(v);
% Normalizar
w=w/sum(w);
return
```

minCuadPond.m

```
function w = minCuadPond(M)

s=size(M);
long= (s(1)*s(2))-s(1);

H2 = zeros([long,s(2)]);

% Generar b
b=zeros([long+1,1]);
b(long+1) = 1;

k=1;
for i= (1:s(1))
    for j=1:s(2)
        if (i ~= j)
            % Generar H
            H2(k,i)=1;
            H2(k,j)=-M(i,j);
            k=k+1;
        end
    end
end
```

```
H = [H2; ones(1, s(1))];

% Resolver con minimos cuadrados
w=H\b;

% Normalizar
w=w/sum(w);

return
```

errores.m

```
function [errorInf, IndexMaxErr, errorFro, errorUno, errorErrRel] = errores(M,w)

% Error
% Construccion W
W=zeros(size(M));
for i = 1:size(w)
    for j = 1:size(w)
        W(i,j) = w(i)/w(j);
    end
end

% Matriz de residuos
R = abs(M-W) ;

% Norma Infinito: residuo maximo y su indice
[errorInf, I] = max(R(:));
[I_row, I_col] = ind2sub(size(R),I);
IndexMaxErr =[I_row, I_col];
% Norma Frobenius
errorFro = norm(R,'fro');

% Norma 1
errorUno = norm(R,1);

% Norma "ErrorRel"
errorErrRel = norm(R./M, 'fro');

% Norma "Aciertos": suma 1 si esta por encima de 1/2 ???

end
```

ic.m

```
function res = ic(M)
%Autovalor dominante
lambda = max(eig(M));
m = length(M);
%Calculo del indice de consistencia
res = (lambda - m)/ (m - 1);
return
```